TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

POURIA BABAHAJIANI

SEMANTIC SEGMENTATION OF OUTDOOR SCENES USING LIDAR CLOUD POINT

Master of Science thesis

Examiner: Professor Hannu Eskola, Professor Moncef Gabbouj and Dr Lixin Fan
Examiner and topic approved in the Natural Science Faculty Council meeting on 30th July 2014

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master Degree Program in Biomedical Engineering
BABAHAJIANI, POURIA: Semantic Segmentation of outdoor Scenes Using LIDAR cloud point

Master of Science Thesis, 57 pages,
March 2014
Major: Medical Informatics
Major: Biomedical Engineering
Examiner: Professor Hannu Eskola, Professor Moncef Gabbouj, Dr Lixin Fan

Keywords: 3D point cloud, LiDAR, classification, segmentation, image alignment, street view, feature extraction, machine learning, Mobile Laser Scanning

In this paper we present a novel street scene semantic recognition framework, which takes advantage of 3D point clouds captured by a high definition LiDAR laser scanner. An important problem in object recognition is the need for sufficient labeled training data to learn robust classifiers. In this paper we show how to significantly reduce the need for manually labeled training data by reduction of scene complexity using non-supervised ground and building segmentation. Our system first automatically segments grounds point cloud, this is because the ground connects almost all other objects and we will use a connect component based algorithm to over segment the point clouds. Then, using binary range image processing building facades will be detected. Remained point cloud will grouped into voxels which are then transformed to super voxels. Local 3D features extracted from super voxels are classified by trained boosted decision trees and labeled with semantic classes e.g. tree, pedestrian, car.

Given labeled 3D points cloud and 2D image with known viewing camera pose, the proposed association module aligned collections of 3D points to the groups of 2D image pixel to parsing 2D cubic images. One noticeable advantage of our method is the robustness to different lighting condition, shadows and city landscape. The proposed method is evaluated both quantitatively and qualitatively on a challenging fixed-position Terrestrial Laser Scanning (TLS) Velodyne data set and Mobile Laser Scanning (MLS), NAVTEQ True databases. Robust scene parsing results are reported.

PREFACE

This Master of Science Thesis has been carried out in the Department of Biomedical Engineering at Tampere University of Technology (TUT), Tampere, Finland as a part of Nokia s project during June 2012 – March 2014. I am pleased to express my gratitude to my thesis supervisors, Professor Hannu Eskola, Professor Moncef Gabbouj and Dr Lixin Fan for their valuable guidance and support throughout the thesis period. I would also like to express my appreciation to my seniors, Viktor Vad and Junsheng Fu and who have worked in the same project. Their way of sharing knowledge and willing to help attitude has supported me a lot to foster my work in the correct direction. I am also grateful to Esin Goldugan, You Yu, Kimmo Roimela from Nokia Company for their valuable comments and suggestions regarding our results.

Finally, it is a pleasure to thank my family for their support and motivation to pursue this Master's degree.

Feb, 2014

Pouria Babahajiani

CONTENTS

.

1. INTRODUCTION

1.1 Introduction

Analysis of 3D spaces comes from the demand to understand the environment surrounding us and to build more and more precise virtual representations of that space. In the last recent decade, as the three dimensional (3D) sensors begun to spread and the commercially available computing capacity has grown big enough to be sufficient for large scale 3D data processing, new methods and applications were born. In recent years, more and more technologies started to appear that heavily rely on these new 3D methods. Such systems have diverse applications at robotic cars, ships and airplanes. In robotics, it is used for perception of the environment, obstacle detection, and avoidance to navigate safely through environments, especially in the case of autonomous vehicles. in the field of geology where high-resolution digital elevation maps generated by airborne and stationary LiDAR helped in detecting subtle topographic features such as river terraces and river channel banks and enabled many novel studies of the physical and chemical processes that shape landscapes. Other object recognition applications include surveillance, industrial inspection, medical imaging, human computer interaction and intelligent vehicle systems.

Automatic urban scene objects recognition refers to the process of segmentation and classification of objects of interest into predefined semantic labels such as building, tree or car etc. This task is often done with a fixed number of object categories, each of which requires a training model for classification scene components. While many techniques for 2 dimensional object recognition have been proposed, the accuracy of these systems is to some extent unsatisfactory because 2D image cues are sensitive to varying imaging conditions such as lighting, shadow etc.

Much work in vision has been devoted to the problem of segmenting and identifying objects in 2D image data. The 3D problem is easier in some ways, as it circumvents the ambiguities induced by the 3D-to-2D projection, but is also harder because it lacks color cues, and deals with data which is often noisy and sparse. The 3D scan segmentation problem has been addressed primarily in the context of detecting known rigid objects for which reliable features can be extracted. The more difficult task of segmenting out object classes or deformable objects from 3D scans requires the ability to handle previously unseen object instances or configurations. This is still an open problem in computer vision, where many approaches assume that the scans have been already segmented into objects.

Three dimensional data can be produced in several ways. It can be generated via sound propagation (sonars), radio wave propagation (radars) and light propagation (CCD devices, LiDAR devices). In this work, we propose a novel automatic scene parsing approach which takes advantage of 3D geometrical features extracted from Light Detection And Ranging (LiDAR) point clouds. A point cloud is a data set with small units of data, each representing a 3D point in the space. A point essentially has at least three information: its 3 coordinates, x, y and z (which of course can be represented in polar coordinate system or Euler angles or latitude, longitude, altitude in geographic data sets, etc). Additionally color, intensity information could also be provided by some devices.

Since such 3D information is invariant to lighting and shadow, as a result, significantly more accurate parsing results are achieved. While a laser scanning or LiDAR system provides a readily available solution for capturing spatial data in a fast, efficient and highly accurate way, the enormous volume of captured data often come with no semantic meanings. Some of these devices output several million data points per second. Efficient, fast methods are needed to filter the significant data out of these streams or high computing power is needed to post-process all this large amount of data. We, therefore, develop techniques that significantly reduce the need for manual labelling of training data and apply the technique to the all data sets. Laser scanning can be divided into three categories, namely, Airborne Laser Scanning (ALS), Terrestrial Laser Scanning (TLS) and Mobile Laser Scanning (MLS). The proposed method is evaluated both quantitatively and qualitatively on a challenging TLS Velodyne data set and MLS NAVTEQ True datasets.

1.2 Problem and Approach

Semantic segmentation, which refers to the process of simultaneously classifying and segmenting objects in a 2D image or 3D scene, is one of the fundamental problems of computer vision. This task is surprisingly difficult. Human beings has the congenital ability that with a short and simple glance at an environment he can identify or categorize objects despite appearance variations due to change in orientation, color, texture, deformation, illumination, and occlusion. The concept of designing a computer vision object recognition pipeline is to recognize objects that we have never seen before. This task is done based on observing and training our pipeline with a set of object. It is a challenging task to develop such a vision system. Some reasons can be attributed to the following factors: limitations of sensor model, noise in the data, clutter, occlusion and self-occlusion. There are also some criteria which make 3D point cloud information not sufficient to understand the whole scene easily. One of the problem is the density of the provided point cloud which is highly inhomogeneous: the further an object from the sensor is, the sparser its 3D scan will be. Also, as the sensor does not see directly down to the ground, each scan has a hole of about 2 meters radius in the center. In addition to noise, moving points can cause hard challenges for the processing algorithms even more. In a realistic street scene, there are number of moving objects: people, vehicles, some vegetation. All the points belonging to these objects change from frame to frame. A false registration can

easily occur when the algorithm falsely detects moving points as significant points and tries to align consecutive point clouds along these moving points.

Multiple scans of an area will also serve to reduce unwanted features in the dataset, such as the presence of cars and pedestrians, which can otherwise be difficult to detect and remove in an automatic modeling pipeline. However, there are many challenges involved in creating an automated solution to this problem. Firstly, Datasets acquired at different times, even using the same acquisition system, will likely sufferer misalignment, which can be made worse by the duration of the scan, the amount of time since a strong GPS fix, or even weather conditions. Secondly, after the overlapping segments of scans have been successfully aligned, we must determine what has changed. Isolating changes can be problematic for a number of reasons. Scan density will not be uniform, either as a result of the scanner being at a different position and orientation with respect to the scanned surface (such as when driving on the other side of a road), or as a result of comparing scans made with different acquisition setups.

Semantic segmentation of urban scene, in general, is defined as the task of locating and labeling objects in a street scene. This task may contains object class recognition which aims at finding and identifying objects that belong to a certain class. For example one classification system may firstly detect and extract ground component of a scene and then makes a classification and segmentation on remained objects such as car, building and trees. Given a point cloud containing one or more objects of interest and a set of labels corresponding to a set of models known to the system, the semantic segmentation system should assign correct labels to regions, or a set of regions, in the point cloud. These systems relies on the idea of generalizing by using a smaller set of objects to classify a larger diverse dataset.

In this research we focus on a hybrid two stage voxel based classification to address the above mentioned challenges. Firstly, we adopt an unsupervised segmentation method to detect and remove dominant ground and buildings from other LiDAR data points, where these two dominant classes often correspond to the majority of point clouds. Secondly, after removing these two classes, we use a pre-trained boosted decision tree classifier to label local feature descriptors extracted from remaining vertical objects in the scene. Our proposed pipeline gives each object a unique identity which enables accurate class recognition. A complete object class classification system is devised that detects, identifies, and localizes potential objects of interest that have not been previously encountered from a given scenes of urban environments.

1.3 Contribution and publication

The contribution of this work are as follows:

- Develop a novel street object recognition method which is robust to different types of LiDAR point clouds acquisition methods.

- A complete scene parsing system is devised and experimentally validated using 3D urban scenes that have been gathered with different type of LiDAR acquisition devices. The steps such as segmentation, cluster extraction, feature extraction, voxelization are generic and adaptable to solve object class recognition problems in different streets with varying landscape
- Proposed two-stage (supervised and non-supervised) classification pipeline which requires only small amount of time for training.
- Propose to use novel geometric features leads to more robust classification results
- Using LiDAR data aligned to image plane leads to segmentation algorithm which is robust to varying imaging condition.
- We propose a novel method to register 3D point cloud to 2D image plane, and by doing so, occluded points from behind the buildings are properly deleted
- We propose to use a novel LiDAR intensity feature for semantic scene parsing, and demonstrate that combining both LiDAR intensity feature and geometric features leads to more robust classification results. Consequently, classifiers trained in one type of city and weather condition is now possible to be applied to a different scene structure with high accuracy

The following publications and patents were written during the project:

I. Babahajiani, Pouria and Fan, Lixin and Gabbouj, Moncef , 'Semantic Parsing of Street Scene Images Using 3D LiDAR Point Cloud, IEEE International Conference on Computer Vision (ICCV), Sydney 2013'

II. P. Babahajiani, L. Fan, M. Gabbouj, Object Recognition in 3D Point Cloud of Urban Street Scene, IEEE Asian Conference on Computer Vision (ACCV), Singapore 2014.

III. Babahajiani Pouria, Fan Lixin, Patent NC86785

2. PERVIOUS WORK

Automatic scene parsing is a traditional computer vision problem. Many successful techniques have used single 2D image appearance information such as color, texture and shape [23, 24]. By using just spatial cues such as surface orientation and vanishing points extracted from single images considerably more robust results are achieved [25]. In order to alleviate sensitiveness to different image capturing conditions, , many efforts have been made to employ 3D scene features derived from single 2D images and thus achieving more accurate object recognition [26]. For instance, when the input data is a video sequence, 3D cues can be extracted using Structure From Motion (SFM) techniques [27]. With the advancement of LiDAR sensors and Global Positioning Systems (GPS), large-scale, accurate and dense point cloud are created and used for 3D scene parsing purpose.

In the past, research related to 3D urban scene analysis had been often performed using 3D point cloud collected by airborne LiDAR for extracting vegetation and building structures [28]. Hernndez and Marcotegui use range images from 3D point clouds in order to extract k-flat zones on the ground and use them as markers for a constrained watershed [29].Recently, classification of urban street objects using data obtained from mobile terrestrial systems has gained much interest because of the increasing demand of realistic 3D models for different objects common in urban era. A crucial processing step is the conversion of the laser scanner point cloud to a voxel data structure, which dramatically reduces the amount of data to process. Yu Zhou and Yao Yu (2012) present a voxel-based approach for object classification from TLS data [30]. Classification using local features and descriptors such as Spin Image [31], Spherical Harmonic Descriptors [32], Heat Kernel Signatures [33], Shape Distributions [34], and 3D SURF feature [35] have also demonstrated successful results to various extent.

Most of these methods follow a similar procedure including filtering and removing noisy points from acquiesced data, feature extraction and segmentation. So based on these steps in each section the related prior art are reviewed. Section 2.1 reviews the mobile laser scanning technology. Section 2.2 describes the point cloud feature extraction prior arts. The pervious researches of the object recognition are reviewed in section 2.3.

2.1 A review of LiDAR technologies

Generally LiDAR system uses a laser beam to derive distances to objects and therewith determine the positions of those objects surface. Nowadays LiDAR systems are capable of taking hundreds of thousands of accurate distance measurements every second. The LiDAR scanners can be stationary or a part of a mobile mapping system where the sensors are mounted on the vehicle, in this case it needs to determine the position when each laser pulse is transmitted and received.

The technology of laser scanning has been reviewed in different literatures [1, 2]. The first of these vehicles, used in 2008 [3], was an early acquisition platform, consisting of LIDAR scanners mounted on a mobile acquisition platform. The vehicle was capable of obtaining scan data of acceptable density at highway speeds. In addition to LiDAR scanner as a core component of mobile mapping systems a survey-grade differential GPS system (DGPS) is used to keep track of the trajectory of the system while scanning, and an inertial measurement unit (IMU) is utilized to monitor smaller and higher frequency changes in acceleration and attitude. Wehr et al. give an introduction about different scanning mechanisms, and various topic related to LiDAR and also principle of airborne laser scanning [4]. The principle and component of LiDAR scanner system is described in section 3 more in detail.

2.2 Feature extraction

This chapter describes the extraction of features for the classification of outdoor-scanned LIDAR data. Here, the term ‘features’ means variables which are extracted from the raw 3D point cloud that are appropriate and distinctive for correct classification with low probability of mismatch. In order to fully parse 3D point cloud, for scene understanding and object classification, effective feature extraction has proved to be a necessary and critical for describing difference among objects since it will be used automatically.

Existing features used for 3D point cloud classification include intensity [5], height [6- 5, 7], surface curvature [6], spin image [7-8], shape distribution [9, 10], local tensors [11], shape maps [12], 3D active contour [13], normal vector [15] and color [14]. These features are often used together, or treated independently as feature descriptors. Furthermore base on desire object classes intended to be determined different features will be used. For example, in order to detect building facades, Pu et al. presented a knowledge based reconstruction of building façade models from terrestrial laser scanned data which takes advantage of combined features e.g. size, position, orientation, and topology and point cloud density [16]. In addition, Armesto- Gonzalez et al. and Dash et al. (2004) developed similar feature based methods to extract information from terrestrial laser scanned data to detect damage and deformations of buildings [17, 18].

Ground segmentation was one of area which were extracted using LiDAR features. For instance Hu et al. did the road extraction from urban area using airborne LiDAR data and high resolution images [19]. Pu et al. developed an automated method for ground segmentation by considering characteristic of features like position, orientation, shape, etc. as well as their topological relations like intersection and angle [20].

Using terrestrial laser scanning for 3D modeling of tree structure is another example which has been done by Rosell et al [21]. The quality of sensor data and the complexity of the target feature will extremely important in object detection procedure. This criteria is by far important in detection of small objects such as tree, pedestrian and sign symbol. Normally, the characteristics of natural objects such as tree is different from the manmade

objects such as cars and buildings. Shape features such as size, area, density, height above ground are used to detect tress [22].

2.3 Segmentation and classification of 3D Data

I present a short overview of the state of the art results on this topic. The literature review presented here is divided into two main sections: segmentation and classification. In each of these sections the relevant work is discussed and grouped under different techniques used in these domains.

2.3.1 Segmentation of 3D point cloud

In order to analyze and apply 3D point clouds for scene understanding and object classification, effective segmentation has proved to be a necessary and critical pre-processing step in a number of autonomous perception tasks.

Rabbani (2006), employed the use of surface discontinuities and small sets of specialized features, such as local point density or height from the ground, to discriminate only few object categories in outdoor scenes, or to separate foreground from background [36]. Moosman et al. [37], investigates about 3D urban scene molding based on surface discontinuities. In the proposed system they used surface convexity in a terrain mesh as a separator between different objects.

Lately, segmentation has been commonly formulated as graph clustering such as Graph-Cuts including Normalized-Cuts and Min-Cuts. The earliest graph-based methods use fixed thresholds and local measures in computing a segmentation. The work of Zahn (1971) presents a segmentation method based on the minimum spanning tree (MST) of the graph [39]. This method has been applied both to point clustering and to image segmentation. For image segmentation the edge weights in the graph are based on the differences between pixel intensities, whereas for point clustering the weights are based on distances between points. Golovinskiy and Funkhouser [38] extended Graph-Cuts segmentation to point clouds by using k-Nearest Neighbors (k-NN) to build a 3D graph. They used edge weights based on exponential decay in length. The result of this work is acceptable however the limitation of this method is that it requires prior knowledge of the location of the objects to be segmented. Zhu et al. [40] presented a method in which a 3D graph is built with k-NN while assuming the ground to be flat for removal during pre-processing. We have used the same assumption. Strom et al. [41] proposed a modified FH algorithm (Felzenszwalb and Huttenlocher) to incorporate angle differences between surface normal in addition to the differences in color values. Segmentation evaluation was done visually without ground truth data. Our approach differs from the abovementioned methods as, instead of using the properties of each point for segmentation resulting in over segmentation, we have grouped the 3D points based on similarity into voxels and then assigned normalized properties to these voxels. This not only prevents over segmentation but in fact reduces the data set by many folds thus reducing post-processing time.

In the literature review, we also find some techniques that segment and model a point cloud through a graph-based ellipsoidal region growing process. A spanning tree approach to the segmentation of 3D point clouds was proposed in [42]. At the heart of the method is a minimum spanning-tree (MST) implementation which grows ellipsoidal segments from initial ellipsoids as the tree expands. The resulting segmentation is similar to a super-voxel type of partitioning with voxels of ellipsoidal shapes and various sizes.

Another set of approaches such as [43, 44], segment and label 3D points by employing Markov Random Fields to model their relationship in the local vicinity. The MRF models incorporate a large set of diverse features and enforce the preference that adjacent scan points have the same classification label. These techniques proved to outperform classifiers based only on local features, but at a cost of computational time.

2.3.2 Classification of 3D Data

In the past, research related to 3D urban scene classification and analysis had been mostly performed using either 3D data collected by airborne LiDAR for extracting bare-earth and building structures [45, 46] or 3D data collected from static terrestrial laser scanners for extraction of building features such as walls and windows [47]. Recently, classification of urban environment using data obtained from mobile terrestrial platforms (such as [48]) has gained much interest in the scientific community due to the ever increasing demand of realistic 3D models for different popular applications coupled with the recent advancements in the 3D data acquisition technology.

The existing work on the problem in the context of 3D scan data classification can largely be classified into three groups. The first class of methods performs classification of 3D shapes. Some methods (particularly those used for retrieval of 3D models from large databases) use global shape descriptors [49, 50], which require that a complete surface model of the query object is available. Objects can also be classified by looking at salient parts of the object surface [51, 52]. All mentioned approaches assume that the surface has already been pre-segmented from the scene. Another line of work performs segmentation of 3D scans into a set of predefined parametric shapes. Han et al. [53] present a method based for segmenting 3D images into 5 parametric models such as planar, conic and B-spline surfaces. Unlike their approach, ours, as third method, is aimed at learning to segment the data directly into objects or classes of objects. This method aims to detect known objects in the scene. Such approaches center on computing efficient descriptors of the object shape. Local 3D geometrical features extracted from subsets of point clouds are classified by classifier such as SVM or boosted decision tree.

Classification using global features is presented in [54] in which a single global spin image for every object is used to detect cars in the scene, while in [55] a Fast Point Feature Histogram (FPFH) local feature is modified into global feature for simultaneous object identification and view-point detection. Classification using local features and descriptors such as Spin Image, Spherical Harmonic Descriptors, Heat Kernel Signatures, and Shape

Distributions is also found in the literature survey. There is also a third type of classification based on Bag Of Features (BOF) as discussed in [56].

In [57] the authors propose using multi-scale Conditional Random Fields to classify 3D outdoor terrestrial laser scanned data by introducing regional edge potentials in addition to the local edge and node potentials in the multi-scale Conditional Random Fields. This is followed by fitting Plane patches onto the labeled objects such as building terrain and floor data using the RANSAC algorithm as a post-processing step to geometrically model the scene. In [58] the authors extracted roads and objects just around the roads like road signs. They used a least square fit plane and RANSAC method to first extract a plane from the points followed by a Kalman filter to extract roads in an urban environment. Douillard et al. [59] presented a method in which 3D points are projected onto the image to find regions of interest for classification. In our work, such as Douillard, we use 2D-3D association to extract geometrical as well as reflection properties of point cloud to successfully classify different segmented objects represented by groups of voxels in the urban scene.

Classifiers always play the last stage of the object detection researches. Feature vectors are labeled using classifier. The labeled object will be determined and each associated with a set of points. The several classifier such as K-neighbors (NN), support vector machine (SVM) and Boosted decision tree are used in the classification procedure. Golovinskiy et al [60] labeled feature vectors by SVM, trained on a set of manually labeled objects (Groundtruth). The idea of segment based classification approach for object detection is introduced by Khoshelham and Elberink (2012) [61], where feature vectors are extracted for each segments and used for the final classification steps. The quantity of training samples and features as well as complexity of classifiers will be firmly effect on the classification results [62]. In our work, local 3D geometrical features extracted from subsets of point clouds are classified by trained boosted decision trees and then corresponding image segments are labeled with semantic classes e.g. buildings, road, sky etc

3. MATERIAL

In this chapter, I introduce the working principles of the Light Detection and Ranging (LiDAR) technology and show examples of devices and their usage. Mathematical and physical formulation for direct georeferencing and technologies applied for the determination of Mobile Laser Scanning (MLS) trajectory are introduced. Furthermore, the different coordinate systems and transformations between them are discussed.

3.1 A review of LiDAR Technology

The primary technology behind this research is the remote sensing technology known as LIDAR (Light Detection and Ranging). The principals of LiDAR distance measurement is essentially the same as of a radar device but instead of radio waves, a LiDAR uses light to measure distances, thus the name Light Detection and Ranging. LIDAR is a fast technology for sampling object surfaces with high density and high accuracy.

A big advantage of this technology over conventional optical imagery, that it is not affected by lighting conditions, so lack of external lightning (e.g.: at night) does not corrupt the measurement. Also, special types of LiDARs are able to scan through water, thus being able to scan underwater surfaces. The pros and cons of both LIDAR and photogrammetry (see table 3.1 and 3.2) and the complementary nature of such characteristics continuously push towards the integration of both systems. Such integration would lead to a more complete surface description from semantic and geometric points of view.

Table 1. *Photogrammetric weaknesses as contrasted by LIDAR strengths.*

LIDAR Pros	**Photogrammetric Cons**
Dense geo-reference information from homogeneous surfaces	Almost no positional information along homogeneous surfaces
Day or night data collection	Day time data collection
Direct acquisition of 3D coordinates	Complicated and sometimes unreliable matching procedures
Robust for shadow and light condition	Dependent on lighting condition

Table 2. *LIDAR weaknesses as contrasted by Photogrammetric strengths.*

Photogrammetric Pros	**LiDAR Cons**
High redundancy	No inherent redundancy
Easy to visually interpretation	Hard to interpret complicated object
Rich in semantic information	difficult to derive semantic information
cheap	expensive

The selection of an imaging sensor is dependent on the desired accuracy, reliability, operational flexibility and application requirements. Digital frame cameras were generally developed for terrestrial based mapping. However, with the steady increase in spatial resolution of digital cameras, these cameras are now being used in airborne applications. Digital cameras are used to capture images using either a Charge Coupled Device (CCD) or a Complementary Metal Oxide Semiconductor (CMOS) system which converts acquired radiation into a charged signal.

The quality of the final synergic product definitely depends on the quality achieved from each individual LiDAR and camera system and the way they are aligned. With calibration, the use of a multi-sensor system (laser scanner and camera) permits more complete and efficient data acquisition. This multi-sensor system provides a high resolution and complete coverage of the environment for urban modelling.

LIDAR enables remote sensing by measuring the Time Of Flight (TOF) of a laser pulse, and using this measurement to determine the distance of the object of which the laser is reflected. The distance of an object is calculated from the time it took the light beam to bounce and arrive back from and object. Since we know the speed of light, the total distance traveled is simply given by the time multiplied by the speed of light.

$$object\ distance = \frac{time\ of\ flight * speed\ of\ light}{2} \tag{3.1}$$

The time interval of the reflected pulse can be also determined based on phase shift ranging method. In this method, the laser scanning system measures the phase difference between the transmitted and reflected pulse.

$$T = \lambda\,(n + \xi/2\pi) \tag{3.2}$$

Where T is the time interval, ξ is the phase difference, λ is the wavelength of the pulse and n is the integer number which are measured using a digital pulse counting technique.

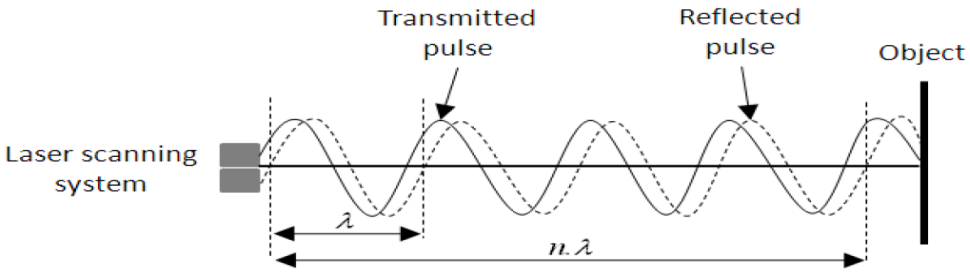


Figure 1. Phase shift method.

The range is estimated from the measured time interval using the distance time relation described in Equation 3.1. These equations are only an approximation; real-world systems adjust for error based on the intensity of the measured pulse returned, environmental and atmospheric conditions, distance to the scanned object, and many more circumstances which should be considered.

The method calculates the distance based on the phase of a returning pulse and in a coordinate system fixed to the laser scanner, not an absolute coordinate system fixed to the earth. So the acquired data is defined with respect to the position and orientation of the scanner. As the acquisition platforms are mobile, the coordinate system is constantly changing and these measurements are difficult to use in their raw format; to simplify analysis of the LiDAR data, scan points are projected into a fixed coordinate system. For our data, this is a geodetic coordinate system, transforming each measurement into a tuple consisting of Latitude, Longitude, and Altitude (WGS84 coordinates).

This conversion is accomplished via a fusion process that correlates the raw range data with information from other sensors on the vehicle. Mobile laser scanning systems typically rely on combination of Global Navigation Satellite Systems (GNSS) and Inertial Measurement Unit (IMU) technology for direct geo - referencing of the data. The primary instrument is a GNSS receiver, such as Global Positioning System (GPS) provide accurate positioning at low data frequency, typically 1-10 observations a second, when satellite visibility and constellation geometry is adequate. The GPS system accuracy can be compromised under a number of conditions, such as dense urban coverage, unfavorable GPS satellite configurations, and weather conditions. The acquisition vehicles also use an inertial measurement unit (IMU) in conjunction with wheel sensors to reduce drift in the GPS data. The IMU sensor measures micro-scale accelerations and changes in attitude at approximately 2 KHz. This data is used to interpolate how the system is moving in space (i.e., change in X, Y, Z and pitch, yaw, roll) between carrier-phase GPS measurements. Additionally, vehicles are equipped with a camera array that periodically obtains a color panorama of the surrounding environment. This panorama data can be correlated with the other sensory information.

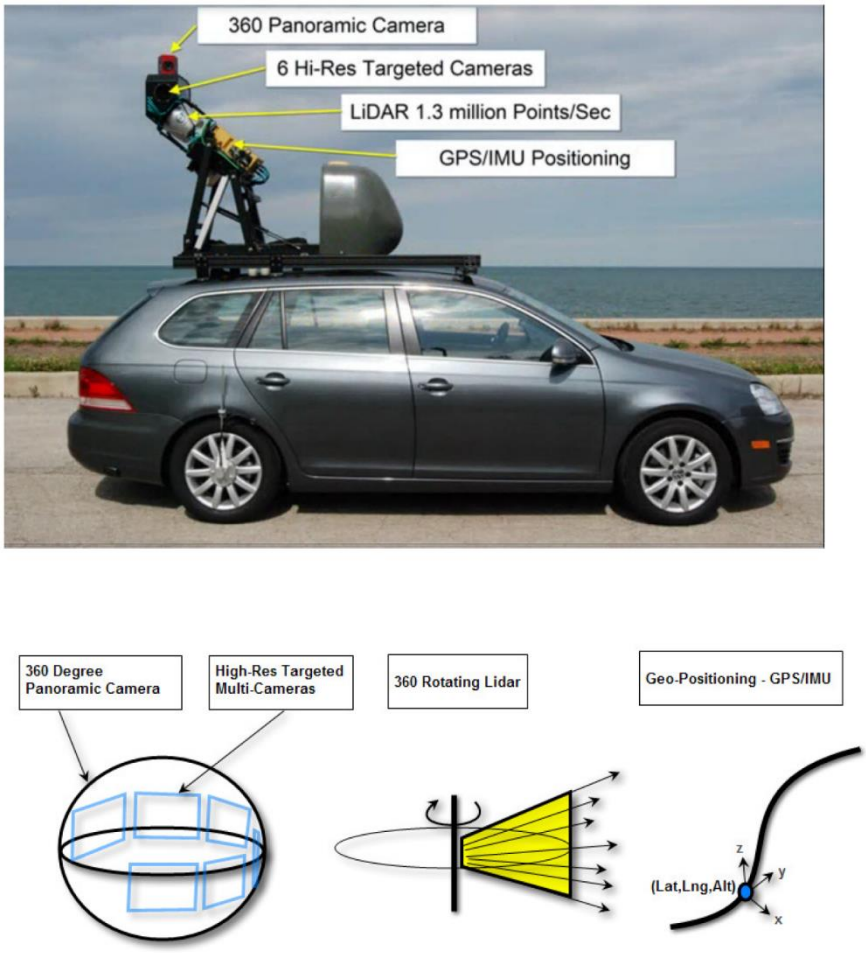


Figure 2. *Integrated Multi-sensor Collection Vehicle [22]*

Different LiDAR vehicles have different operation principles, however following components are mostly common and have to work simultaneously for the generation of a precise digital surface model (see figure 1):

1. 360 panoramic camera: the imaging system along with laser scanning
2. Laser Range Finder (LRF): Measures the distance very accurately. It comprises the laser, transmission and receiving optics, the signal detector, the amplifier and the time counter.
3. Scanner: Deflects the laser beam across the path.
4. Global Positioning System (GPS): Determines the position and path of the vehicle using differential GPS positioning.
5. Inertial Measurement Unit (IMU): Measures acceleration and attitude changes and integrates them.

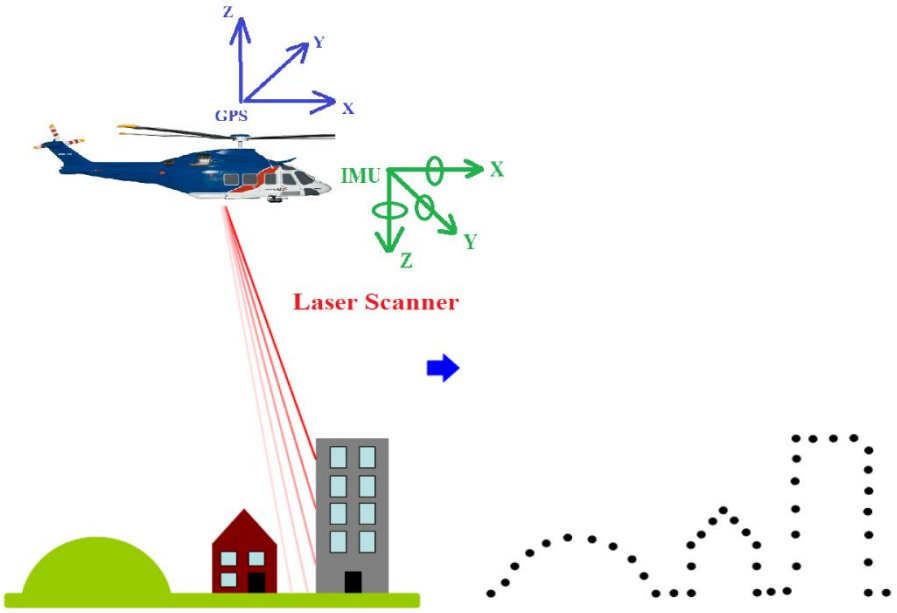


Figure 3. *ALS data acquisition*

3.2 Types of LiDAR Systems

There are two types of urban LiDAR data acquisition in general: airborne and terrestrial. The urban 3D data can be scanned with airborne laser scanners. In this case the airborne laser scanner is generally handheld or mounted on a helicopter. Data acquisition for a terrestrial LiDAR system, on the other hand, is via a close-range terrestrial laser scanner mounted on a mobile robot or vehicle. There is a statistic terrestrial LiDAR which are typically used to capture features of interest in high detail. To capture large features such as buildings often requires taking multiple scans from different locations. Static scanners have the advantage of precision and affordability.

3.2.1 Airborne LiDAR

The Airborne laser scanning is an efficient system which can deliver very dense and accurate point clouds from the ground surface and the objects which are located on it. Providing high quality height information of the landscape by means of LIDAR systems opens up an extensive range of applications in different subjects in photogrammetry and remote sensing. The majority of change detection work in the context of Geographic Information System (GIS) is based on data obtained aerially. For urban applications, many research projects first make segmentation, and then attempt to monitor how segments change over time.

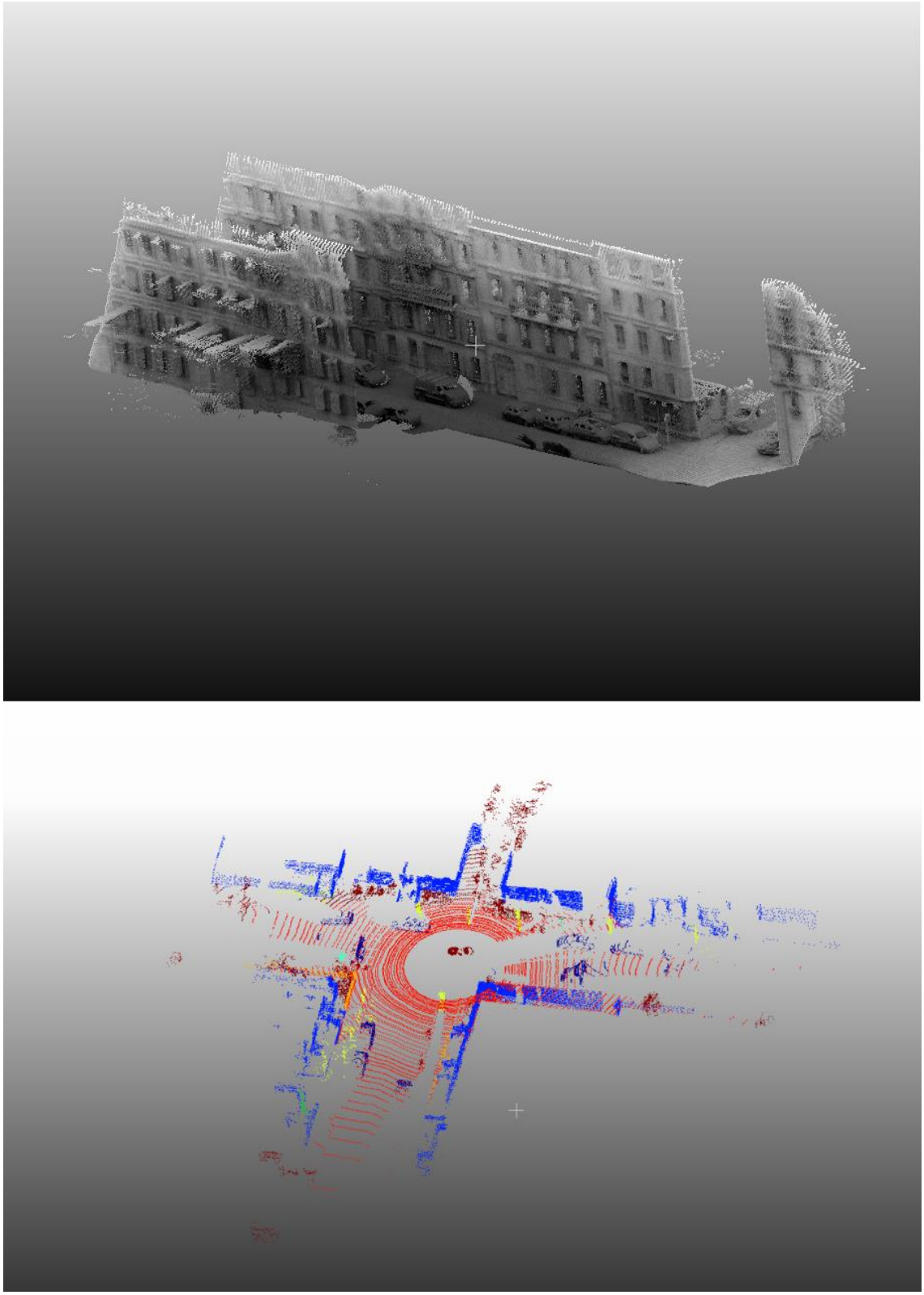


***Figure 4**. Example scanned and filtered MLS data from the NAVTEQ True system*
Example scan MLS data from the Velodyne system

Aerial LiDAR systems typically use a single scanner that has an oscillating or rotating mirror to capture surfaces in a narrow swath below the aircraft. As discussed in LiDAR system components, airborne LiDAR consists of two main systems; Global Positioning System (GPS) and Inertial Navigation System (INS). Figure 3 identifies the main components of the system; laser scanner that provides range and intensity data, GPS that provides 3D positioning and timing parameters, while Inertial Measurement Unit (IMU) that provides orientation parameters.

3.2.2 Mobile Terrestrial LiDAR

The aerial and terrestrial Laser scanning systems which are used to acquire LiDAR data have different and specific usages in the computer vision applications. The data acquired from these systems differs in terms of its intrinsic accuracy and resolution for a variety of reasons but primarily due to the distance of the scanner to the target objects. In recent years, the use of terrestrial based moving vehicles has increased for the collection of high quality 3D data. Terrestrial laser scanning systems have utility towards accurate three-dimensional mapping of urban furniture (e.g. street signs, traffic lights, post boxes, traffic barriers, etc.), road details and vegetation. MLS provides rapid and dense capturing of 3D data for large street sections. The typical MTL vehicle is presented in figure 2 and an example of point cloud from MLS system is shown in figure 4. Terrestrial Laser Scanning (TLS) is a type of ground based LiDAR collecting which in it the LiDAR scanner and other sensors are stationary. In other word, MLS LiDAR is a type of TLS which the equipment are embedded on the top of a moving vehicle.

The ground based mobile LiDAR collection presents many opportunities, but it also presents several challenges compared to aerial LiDAR mapping. A major advantage of ground based mobile LiDAR is that it allows a high-density, focused data collection along the targeted road path. Mobile collection enables true 3D data collection from multiple angles of access, except the roof view. In addition, multiple sensors such as panoramic cameras enable alignment and processing 2D and 3D word together. Another advantage of MLS is that it removes the need to close transportation corridors during data acquisition, to put people in harm's way, or to spend large sums of money on traditional surveying. For example, the vehicle can drive with traffic, the operators are within the vehicle, and the system can make tens of thousands of accurate measurements per second. The challenge is in using these measurements effectively.

The volume and structure of this data presents several challenges. The density of the 3D data collected by TLS scanners is due to the close proximity of the sensor to the targets. In addition, for large scale mapping, the collection vehicle may collect data spanning several hundred kilometers in a single day and for ideal data inspection close 3D viewing at the ground level and in the meter range is essential. Therefore, high techniques to save and display the datasets are needed. In addition, to generate usable geographic information based on unorganized point many challenges which could not be solved in some cases are listed below:

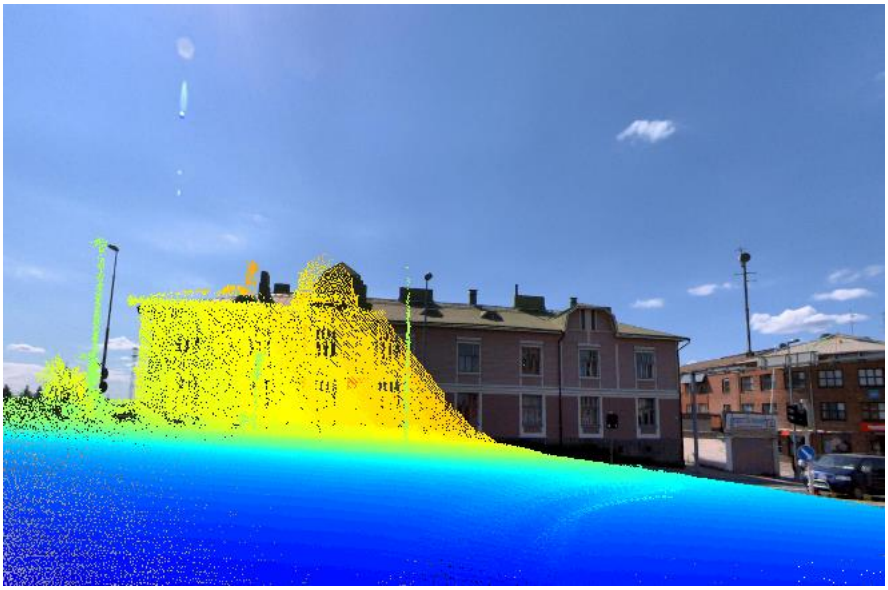


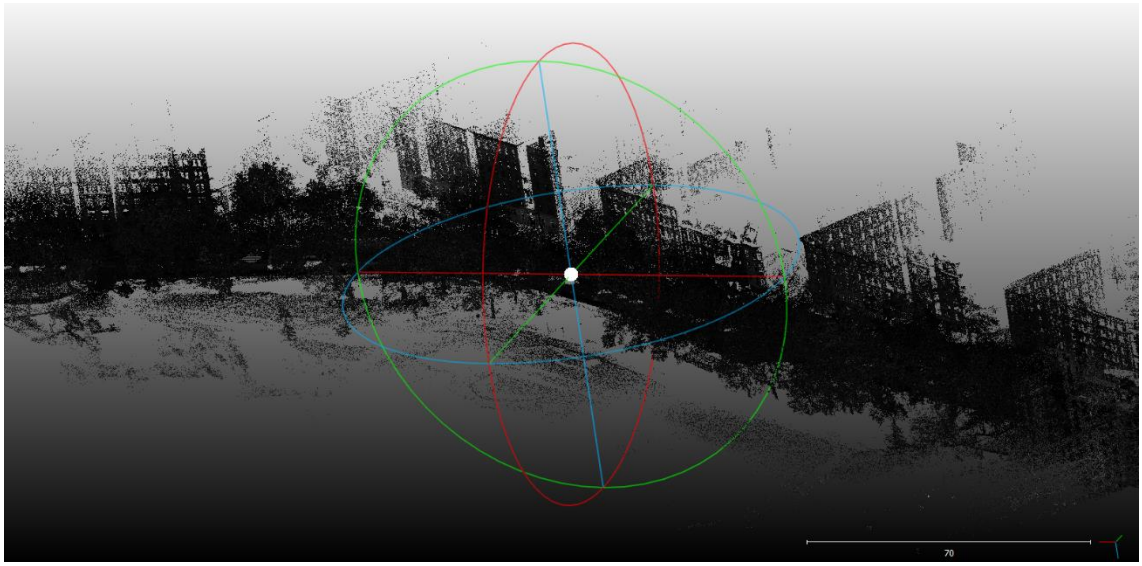
Figure 5. *Projection image and 3D LiDAR point cloud, presents the object boundaries misalignment*

- Objects in the real street scene move while they are being scanned by MLS scanners, and the sensor itself is moving
- Sampling density varies with speed of vehicle acquisition system and surface geometry
- Points rarely fall along object boundaries, it is so important especially in detecting thin objects such as pedestrians and sign symbols, see figure 5
- Foreground objects obstruct the scanners view of interest

NAVTEQ True:

For the experiments, we made the algorithm automatically run through NAVTAQ true datasets provided by HERE. NAVTEQ mobile mapping system called NAVTEQ True, consists of best of sensors in three categories - positioning sensors, LiDAR sensors and imaging sensors. NAVTEQ was an American Chicago-based provider of geographic information systems data and a major provider of base electronic navigable maps. The company was acquired by Nokia in 2007/2008, and fully merged into Nokia in 2011 to form part of the HERE business unit. The sensors include a 360 panoramic camera for a nearly complete spherical view; multiple high resolution cameras for targeted view directions; a high density 360 rotating LiDAR system; and an inertial navigation system (IMU/GPS) for precise position and attitude tracking of the sensors. Information from all these sensors is synchronized to create an accurate and comprehensive data set that can be used for creation of accurate digital maps. The positioning system uses a high accuracy IMU, wheel sensors to measure the distance traveled and an L1/L2 GPS receiver. The LiDAR system has 64 lasers and rotates at 600 rpm covering a 360 degree field of view around the vehicle. The LiDAR system collects three dimensional point cloud at the rate of 1.3

million points per second. The LiDAR sensor returns the X, Y, Z coordinates and the reflectivity of the object that it scans. There are two types of imaging systems used in this mobile mapping vehicle. A panoramic camera that gives 360 degree view of the images around the vehicle and a set of high resolution images targeted at objects of interest. The point cloud data from the LiDAR system and the panoramic images are synchronized to create the colorized 3D point cloud. The picture below shows a sample of the point cloud data gathered using NAVTAQ vehicle system.



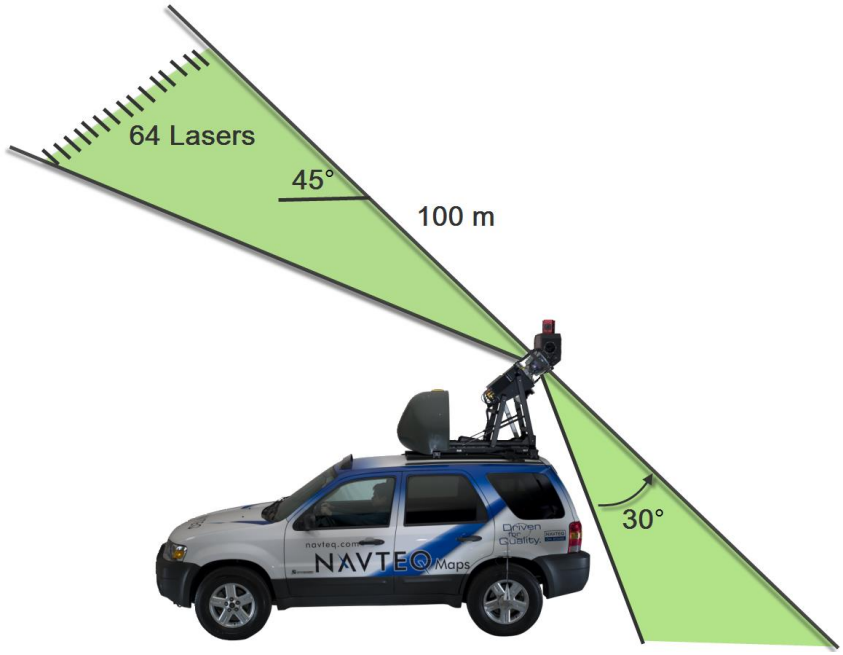


Figure 6. *3D point cloud in its noisy raw original format,*

Data collection vehicle "NAVTEQ True

3.3 Software used for programming and visualization

MATLAB and C++ were the two programming language used for this research. Matlab was mainly used due to its useful tools for image processing and visualization. Furthermore Cloudcompare and Meshlab are used for the purpose of data visualization. Some Point cloud Library (PCL) tools used during our pipeline implementation. For details on this software library, please refer to [63]. Point Cloud Library is a stand-alone, large-scale, open project for 2D/3D image and point cloud processing. The library is freely available under a BSD license and the project is actively maintained through the collaboration of many universities and the industry. The library also defines a widely used file format for storing point clouds. It is a simple format containing a header and the 3D information, functions to read and write this format are available in the library.

4. METHODOLOGY

It is a challenging task to directly extract objects from mobile LiDAR point cloud because of the noise in the data, huge data volume and movement of objects. We therefore take a hybrid two-stage approach to address the above mentioned challenges. Firstly, we adopt an unsupervised segmentation method to detect and remove dominant ground and buildings from other LiDAR data points, where these two dominant classes often correspond to the majority of point clouds. Secondly, after removing these two classes, we use a pre-trained boosted decision tree classifier to label local feature descriptors extracted from remaining vertical objects in the scene. This work shows that the combination of unsupervised segmentation and supervised classifiers provides a good trade-off between efficiency and accuracy. The output of classification phase is 3D labeled point cloud and each point is labeled with a predefined semantic classes such as building, tree, pedestrian and etc.

Given a labeled 3D point cloud and 2D cubic images with known viewing camera pose, the association module aims to establish correspondences between collections of labeled 3D points and groups of 2D image pixels. Every collection of 3D points is assumed to be sampled from a visible planar 3D object i.e. patch and corresponding 2D projections are confined within a homogenous region i.e. SuperPixels (SPs) of the image. The output of the 2D-3D alignment phase is 2D segmented image, in which every pixel is labeled based on 3D cues. In contrast to existing image-based scene parsing approaches, the proposed 3D LiDAR point cloud based approach is robust to varying imaging conditions such as lighting and urban structures.

The work flow which is used to achieve the objective of this research will be elaborated sequentially. As mentioned in chapter 1, the main process as well as the expected result will be discussed in section 4.1. After the proposed mythology workflow given, the detailed analysis for each process will be discussed in the following sections. The proposed method is evaluated both quantitatively and qualitatively and robust scene parsing results are reported in section 5.

4.1 Framework of the methodology

The framework of the proposed mythology is given in figure 7, in which 3D LiDAR point cloud and cubic images are the inputs of the processing pipeline and parsing results are presented as 3D labeled point cloud and 2D image segmented with different class labels e.g. Building, road, car and etc.

There are five main phases in the research: ground segmentation, building detection, voxelization, feature extraction-classification and 2D-3D association. Each phases will be discussed in detail in the following section.

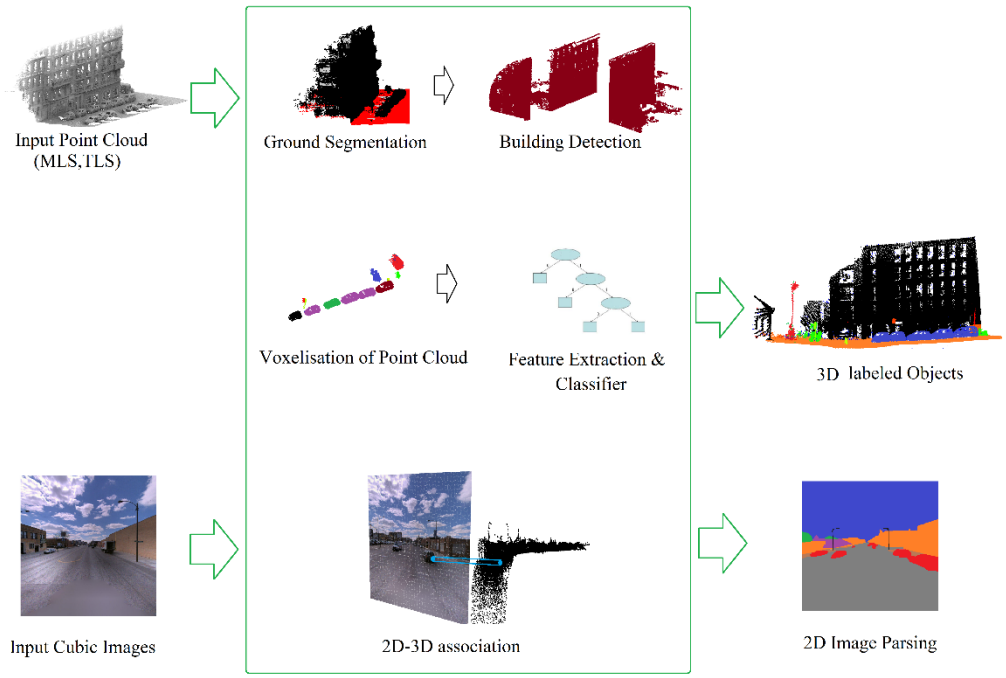


Figure 7. Framework of proposed mythology

Figure 7 shows the overview of the proposed street scene object recognition pipeline, in which LiDAR point cloud and cubic images are the input of the processing pipeline and results are 3D PC and image segments assigned with different class labels. Because 3D cues are more robust compare to image features whole LiDAR data processing, segmentation, feature extraction and classification are done in 3D world.

At the outset, the proposed parsing pipeline finds ground points by fitting a ground plane to the given 3D point cloud of urban street scene. Then, non-ground point cloud are projected to range images because they are convenient structure for visualization. Remaining data are processed subsequently to segment building facades. When this process is completed, range images are projected to the 3D point cloud in order to make segmentation on other remained vertical objects. We use a connect component based algorithm to voxelization of data. The voxel based classification method consists of three steps, namely, a) voxelization of point cloud, b) merging of voxels into super-voxels and c) the supervised scene classification based on discriminative features extracted from super-voxels. Using a trained boosted decision tree classifier, each 3D feature vector is then designated with a semantic label such as tree, car, pedestrian etc. The offline training of the classifier is based on a set of 3D features, which are associated with manually labeled super-voxels in training point cloud. In the last phase 3D labeled PC are used to generate 2D image parsing result. Every images are segmented into superpixels to reduce computational complexity and to maintain sharp class boundaries. Each superpixel in 2D image is associated with a class label based on labeled 3D patch.

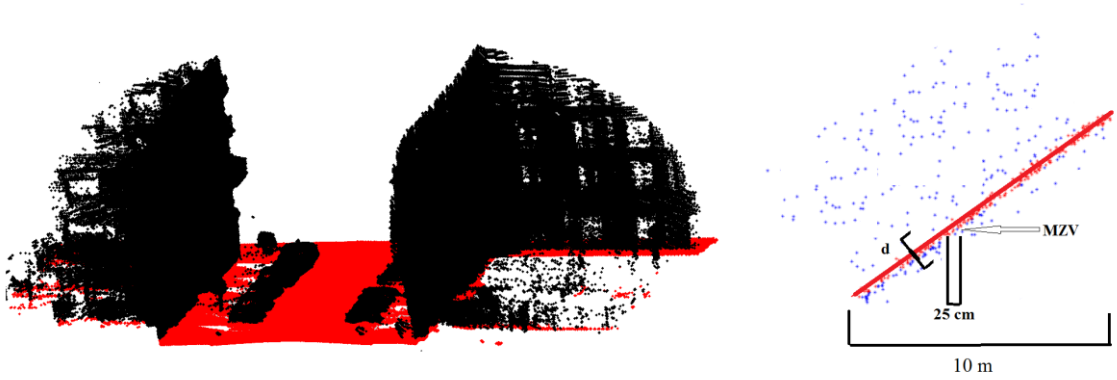


Figure 8. *Ground Segmentation. Left image: Segmented ground and remained vertical objects point cloud are illustrated by red and black color respectively. Right figure: sketch map of fitting plane to one tile*

4.2 Ground segmentation

The aim of the first step is to remove points belonging to the scene ground including road and sidewalks, and as a result, the original point cloud are divided into ground and vertical object point clouds (Figure 8). Given a 3D point cloud of an urban street scene, the proposed approach starts by finding ground points by fitting a ground plane to the scene. This is because the ground connects almost all other objects and we will use a connect component based algorithm to over-segment the point clouds in the following step.

The plane RANSAC fitting method is used to approximate ground section of the scene. The RANSAC algorithm was developed by Fischler et al. [64] and is used to provide a more robust fitting of a model to input data in the presence of data outliers. Unlike conventional model fitting techniques that use as much data as possible to obtain an initial solution, the RANSAC algorithm uses the smallest set of initial data required to fit a model and enlarges this set with compatible data. If there are enough compatible data, RANSAC can improve the estimation of the model, without having to deal with the data outliers. We will now describe the RANSAC algorithm in detail.

Suppose that we have n points in a dataset, $X = x_1, x_2, \ldots x_n$. A minimum required number of m points are randomly selected, such that $m \leq n$, to fit a least-square model M. The least-square model is fitted to the points based on minimizing the sum of square residuals which are the difference between the actual points and the fitted points. The model M is used to estimate data points in X (consensus points) which are within an error tolerance parameter, ε. If the number of consensus points is equal to or larger than a threshold, t, then a new least square model M* is fitted to these points. Otherwise, the whole process is repeated beginning with a random selection of m points. After some pre-set number of iterations, K, if the number of consensus points equal to or larger than t is not found, then

either the model fitted with the largest number of consensus points is accepted or the process is terminated unsuccessfully.

Given a 3D point cloud of an urban street scene, the scene point cloud is first divided into sets of 10m×10m regular, non-overlapping tiles along the horizontal x–y plane. Then the following ground plane fitting method is repeatedly applied to each tile. We assume that ground points are of relatively small z values as compared to points belonging to other objects such as buildings or trees (see Figure 8). The ground is not necessarily horizontal, yet we assume that there is a constant slope of the ground within each tile. Therefore, we first find the minimal-z-value (MZV) points within a multitude of 25cm×25cm grid cells at different locations. For each cell, neighboring points that are within a z-distance threshold from the MZV point are retained as candidate ground points. Subsequently, a RANSAC method is adopted to fit a plane to candidate ground points that are collected from all cells. The RANSAC algorithm uses three specified parameters as ε= 0.05 m, t = 7000 and K = 50. Finally, 3D points that are within certain distance (d in Figure 8) from the fitted plane are considered as ground points of each tile. The constant slope assumption made in this approach is valid for our data sets as demonstrated by experimental results in Section 5.

The approach is fully automatic and the change of two thresholds parameters do not lead to dramatic change in the results. On the other hand, the setting of grid cell size as 25cm×25cm maintains a good balance between accuracy and computational complexity.

4.3 Building segmentation

After segmenting out the ground points from the scene, we present an approach for automatic building surface detection. High volume of 3D data impose serious challenge to the extraction of building facades. Our method automatically extract building point cloud (e.g. doors, walls, facades, noisy scanned inner environment of building) based on two assumptions: a) building facades are the highest vertical structures in the street; and b) other non-building objects are located on the ground between two sides of street.

As can be seen in figure 9, our method projects 3D point clouds to range images because they are convenient structures to process data. Range images are generated by projecting 3D points to horizontal x–y plane. In this way, several points are projected on the same range image pixel. We count the number of points that falls into each pixel and assign this number as a pixel intensity value. In addition, we select and store the maximal height among all projected points on the same pixel as height value. We define range images by making threshold and binarization of I, where I pixel value is defined as equation 4.1

$$I_i = \frac{P_{intensity}}{\text{Max\_}P_{intensity}} + \frac{P_{\text{height}}}{\text{Max\_}P_{\text{height}}} \qquad (4.1)$$

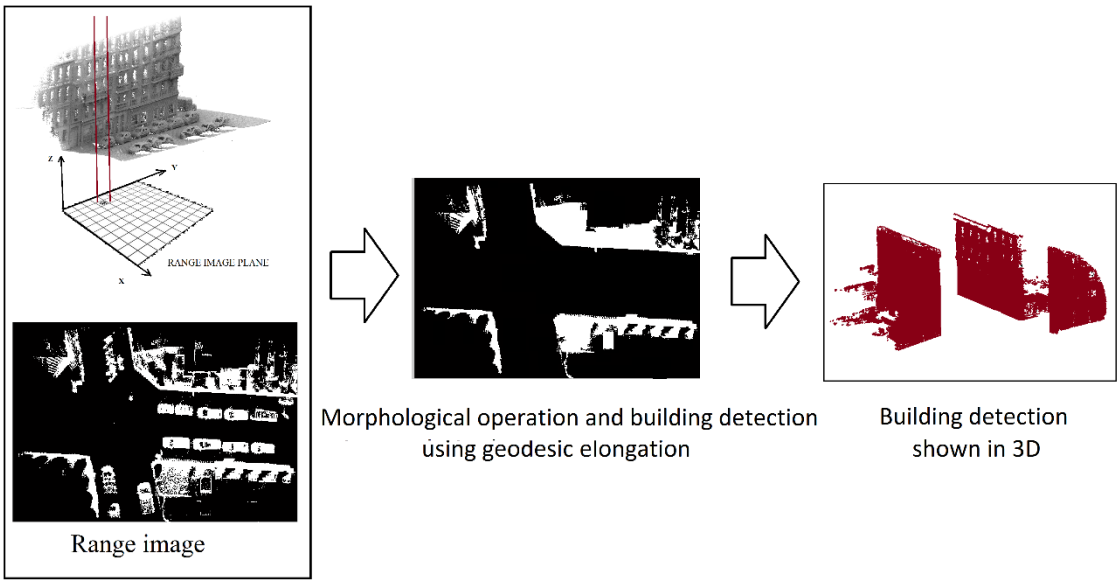


Figure 9. *Building Segmentation*

Where I_i is grayscale range image pixel value, $P_{intensity}$ and P_{height} are intensity and height pixel value and $Max\_P_{intensity}$ and Max_P_{height} represent the maximum intensity and height value over the grayscale image. On the range image, an interpolation is required in order to fill holes caused by occlusions, missing scan lines and LiDAR back projection scatter.

In the next step we use morphological operation (e.g. close and erode) to merge neighboring point and filling holes in the binary range images (see middle image in Figure9). The morphological interpolation does not create new regional maxima, furthermore it can fill holes of any size and no parameters are required. Then we extract contours to find boundaries of objects. In order to trace contours, Pavlidis contour-tracing algorithm [65] is proposed to identify each contour as a sequence of edge points. The resulting segments are checked on aspects such as size and diameters (height and width) to distinguish building from other objects. More specifically, equation (4.2) defines the geodesic elongation E(X), introduced by Lantuejoul and Maisonneuve (1984), of an object X, where S(X) is the area and L(X) is the geodesic diameter.

$$E(\pi) = \frac{\pi L^2(X)}{4S(X)} \qquad (4.2)$$

The compactness of the polygon shape based on equation (4.2) can be applied to distinguish buildings from other objects such as trees. Considering the sizes and shape of buildings, the extracted boundary will be eliminated if its size is less than a threshold. The proposed method takes advantage of priori knowledge about urban scene environment and assumes that there are not any important objects laid on the building facades. While this assumption appears to be oversimplified, the method actually performs quite well with urban scenes as demonstrated in the experimental results (see section 5).

The resolution of range image is the only projection parameter during this point cloud alignment that should be chosen carefully. If each pixel in the range image cover large area in 3D space too many points would be projected as one pixel and fine details would not be preserved. On the other hand, selecting large pixel size compared to real world resolution leads to connectivity problems which would no longer justify the use of range images. In our experiment, a pixel corresponds to a square of size .05 m2.

The 2D image scene is converted back to 3D by extruding it orthogonally to the point cloud space. The x–y pixels coordinate of the binary image labeled as building facades are preserved as x–y coordinate of 3D point cloud (with open z value) labeled as building, and not considered in the remainder of our approach. Other points (negligible amount compare to the size of whole PC) are labeled as non-building class and will be later be classified as other classes e.g. car, tree, pedestrian and etc.

4.4 Voxel based segmentation

After quick segmenting out the ground and building points from the scene, we use an inner street view based algorithm to cluster point clouds. Although top view range image analysis generates a very fast segmentation result, there are a number of limitation to utilize it for the small vertical object such as pedestrian and cars. These limitations are overcome by using inner view (lateral) or ground based system in which, unlike top view the 3D data processing is done more precisely and the point view processing is closer to objects which provides a more detailed sampling of the objects. However, this leads to both advantages and disadvantages when processing the data. The disadvantage of this method includes the demand for more processing power required to handle the increased volume of 3D data.

The 3D point clouds by themselves contain a limited amount of positional information and they do not illustrate color and texture properties of object. According to voxel based segmentation, points which are merely a consequence of a discrete sampling of 3D objects are merged into clusters voxels to represent enough discriminative features to label objects. 3D features such as intensity, area and normal angle are extracted based on these voxels. The voxel based classification method consists of three steps, voxelization of point cloud, merging of voxels into super-voxels and the supervised classification based on discriminative features extracted from super-voxels. In the following two subchapters we present the concept and properties of voxels and supervoxels, and in the next chapters we present the way for extracting discriminative features from these 3D supervoxels.

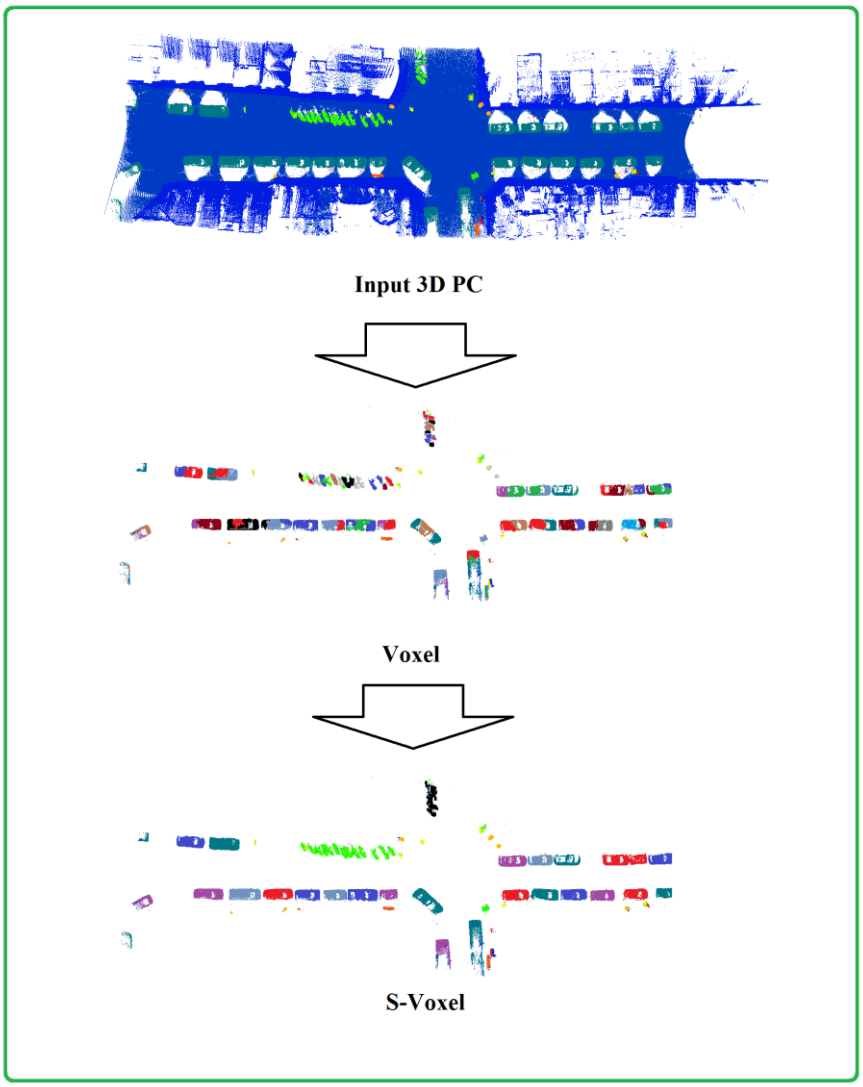


Figure 10. *Voxelization of Point Cloud. from top to down: top view row point cloud, voxelization result of objects point cloud after removing ground and building, s-voxelization approach of point cloud*

4.4.1 Voxelization of Point Cloud

In the voxelization step, an unorganized point cloud *p* is partitioned into small parts, called voxel *v*. The middle image in figure 10 illustrates an example of voxelization results, in which small vertical objects point cloud such as cars are broken into smaller partition. Different voxels are labelled with different colors. The aim of using voxelization is to reduce computation complexity by and to form a higher level representation of point cloud scene.

Following [66], a number of points is grouped together to form a variable size voxels. The criteria of including a new point p_{in} into an existing voxel *i* is essentially determined by the crucial minimal distance threshold d_{th} which is defined as equation (4.3).

$$\min(\|P_{im} - P_{in}\|_2) \leq d_{th}, \quad 0 \leq m, n \leq N, \quad m \neq n \tag{4.3}$$

where p_{im} is an existing 3D point in voxel, p_{in} is a candidate point to merge to the voxel, i is the cluster index, d_{th} is the maximum distance between two point, and N is the maximum point number of a cluster. If the condition is met, the new point is added and the process repeats until no more point that satisfies the condition is found (see Algorithm 1). Equation (4.3) ensures that the distance between one point and its nearest neighbors belonging to the same cluster is less than d_{th}. Although the maximum voxel size is predefined, the actual voxel sizes depend on the maximum number of points in the voxel (N) and minimum distance between the neighboring points.

Repeat

> *Select a 3D point for Voxelization;*
>
> *Find all neighboring points to be included in the voxel, with this condition that:*
>
> *a point p_{in} directly merge to voxel if its distance to any point p_{in} the voxel will not be farther away than a given distance (d_{th});*

Until *all 3D points are used in a voxel or the size of cluster is less than (N)*

Algorithm 1: *Voxelization*

The voxel example of one scene is presented in the middle image of Figure 10. Different voxels are labelled with different colors.

4.4.2 Super Voxelization

For transformation of a voxel to super voxel we propose an algorithm to merge voxels via region growing with respect to the following properties of clusters:

- **If the minimal geometrical distance, D_{ij}, between two voxels is smaller than a given threshold**, where D_{ij} is defined as equation (4.4):

  $$D_{ij} = \min\left(\left\|P_{ik} - P_{jl}\right\|_2\right), k \in (1, m), l \in (1, n) \tag{4.3}$$

  Where voxels v_i and v_j have m and n points respectively, and p_{ik} and p_{jl} are the 3D point belong to voxel v_i and v_j.

- **If the angle between Normal vectors of two voxels is smaller than a threshold**: In this work, normal vector is calculated using PCA (Principal Component Analysis) [67]. The angle between two s-voxels is defined as angle between their normal vectors as equation 4.5:

$$\theta_{ij} = \arccos(< n_i , n_j >) \quad (4.5)$$

Where n_i and n_j are normal vectors at v_i and v_j respectively.

The proposed grouping algorithm merges the voxels by considering the geometrical distance ($D_{ij} < d_{th}$) and normal features of clusters ($\theta_{ij} < \theta_{th1}$). All these Voxelization steps then would be used in grouping these super-voxels (from now onwards referred to as s-voxels) into labeled objects.

The advantage of this approach is that we can now use the reduced number of super voxels instead of using thousands of points in the data set, to obtain similar results for classification. The down image in figure 10 illustrates an example of s-voxelization results, in which different s-voxels are labelled with different colors.

4.5 Feature extraction and classification

For each s-voxel, seven main features are extracted to train the classifier. The seven features are *geometrical shape, height above ground, horizontal distance to center line of street, density, intensity, normal angle and planarity*. While the task of segmentation and classification traditionally relies on the color information alone, using such 3D information has some obvious advantages. Firstly it is invariant to lighting and/or texture variation; secondly it is invariant to camera pose and perspective change (view-independent fashion).

4.5.1 Feature extraction

In order to classify these s-voxels, we assume that the ground points have been segmented well. The object types are so distinctly different however these features as mentioned are sufficient to make a classification. Along with the above mentioned features, geometrical shape descriptors plays an important role in classifying objects. These shape-related features are computed based on the projected bounding box to x - y plane (ground).

Geometrical shape: Projected bounding box has effective features due to the invariant dimension of objects. We extract four feature based on the projected bonding box to represent the geometry shape of objects.

- Area: the area of the bounding box is used for distinguishing large-scale objects and small ones, (see figure 11).
- Edge ratio: the ratio of the long edge and short edge.
- Maximum edge: the maximum edge of bounding box.
- Covariance: is used to find relationships between points spreading along two largest edges.

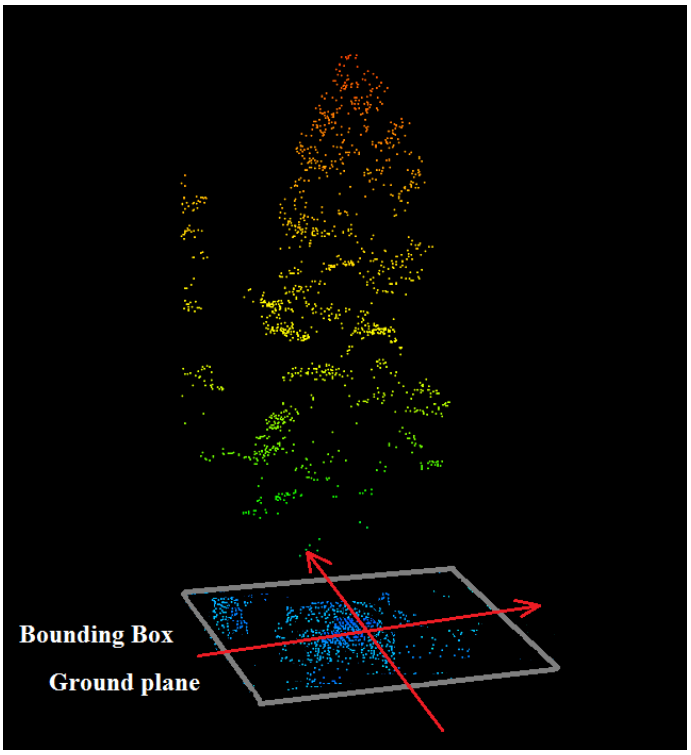


Figure 11. *Bounding box for tree*

Height above ground: Given a collection of 3D points with known geographic coordinates, the median height of all points is considered as the height feature of the s-voxel. The height information is independent of camera pose and is calculated by measuring the distance between points and the road ground.

Horizontal distance to center line of street: Following [68], we compute the horizontal distance of the each s-voxel to the center line of street as second geographical feature. The street line is estimated by fitting a quadratic curve to the segmented ground.

Density: Some objects with porous structure such as fence and car with windows, have lower density of point cloud as compared to others such as trees and vegetation. Therefore, the number of 3D points in a s-voxel is used as a strong cue to distinguish different classes.

Intensity: following [69], LiDAR systems provide not only positioning information but also reflectance property, referred to as intensity, of laser scanned objects. This intensity feature is used in our system, in combination with other features, to classify 3D points. More specifically, the median intensity of points in each s-voxel is used to train the classifier.

Normal angle: Following [70], we adopt a more accurate method to compute the surface normal by fitting a plane to the 3D points in each s-voxel. The surface normal is important properties of a geometric surface, and is frequently used to determine the orientation and general shape of objects. A surface normal is calculated using PCA (Principal Component Analysis). Given 3D point cloud data set $D = x_1, x_2, x_3, \ldots, x_n$, the PCA surface normal

approximation for a given data point p ∈ D is typically computed by first determining the k-Nearest Neighbors (KNN), $x_k \in D$, of p. Given the K neighbors, the approximate surface normal is then the eigenvector associated with the smallest eigenvalue of the symmetric positive semi-definite matrix

$$P = \sum_{k=1}^{K} (x_k - \bar{p})^T (x_k - \bar{p}) \qquad (4.6)$$

Where $\bar{p}$ is the local data centroid, be the medians are mean of three components of all 3D points. A surface normal is estimated for all the points belonging to a voxel and is then associated with that particular voxel. In the experiments, we only estimate the normal direction for regions containing at least five 3D points. For diluted regions without sufficient points for normal estimation, we let this feature value to be 0.5.

Planarity: Patch planarity is defined as the average square distance of all 3D points from the best fitted plane computed by RANSAC algorithm. This feature is useful for distinguishing planar objects with smooth surface like cars form non planar ones such as trees.

4.5.2 Classifier

The Boosted decision tree [71] has demonstrated superior classification accuracy and robustness in many multi-class classification tasks. Acting as weaker learners, decision trees automatically select features that are relevant to the given classification problem. Given different weights of training samples, multiple trees are trained to minimize average classification errors. Subsequently, boosting is done by logistic regression version of Adaboost to achieve higher accuracy with multiple trees combined together. Each decision tree provides a partitioning of the data and outputs a confidence-weighted decision which is the class-conditional log-likelihood ratio for the current weighted distribution.

The classifier training algorithm is given in Table 4.1. In the experiment the initial distribution is defined as proportional to the percentage density of whole point cloud spanned by each s-voxel, reflecting that correct classification of large s-voxel is more important than of small ones. When computing the log-likelihood ratio, we add a small constant ($\frac{1}{2m}$ for m data samples) to the numerator and denominator which helps to prevent overfitting and to stabilize the learning process. We train separate classifiers to distinguish among the whole classes. These are each learned in a one vs. all fashion.

In our experiments, we boost 10 decision trees each of which has 6 leaf nodes. This parameter setting is similar to those in [72], but with slightly more leaf nodes since we have more classes to label. The number of training samples depends on different experimental settings, which are elaborated in Section 5.

Table 4.1. BOOSTED DECISION TREES

Input:

- $D_1 ... D_m$: training data
- $w_{1,1} ... w_{1,m:}$ initial weights
- $y_1 ... y_m \in \{-1, 1\}$: labels
- n_n: number of nodes per decision tree
- n_t: number of weak learner decision trees

For t = 1...n_t:

I. Learn n_n-node decision tree T_t based on weighted distribution w_t

II. Assign to each node $T_{t,k}$: $f_{t,k} = \frac{1}{2} \log \frac{\sum_{i:y_i=1, D_i \in T_{t,k}} w_{t,i}}{\sum_{i:y_i=-1, D_i \in T_{t,k}} w_{t,i}}$

III. Update weights: $w_{t+1,i} = \frac{1}{1+\exp(y_i \sum_{t'}^{t} f_{t',k_{t'}})}$, with $K_{t':} D_i \in T_{t',kt'}$

IV. Normalize weights so that $\sum_i w_{t+1,i} = 1$

Output:

- $T_1 ... T_{nt}$: decision trees
- $f_{1,1} .. f_{nt,nn}$: weighted log-ratio for each node of each tree

For instance, to distinguish among the 10 defined classes, we train 10 classifiers that estimate the probability of a s-voxel being car, pedestrian, tree or etc. These are then normalized to ensure that the estimated probabilities sum to one.

To train the classifiers, we need to assign ground truth to the automatically created S-voxel. If nearly all (at least 90%) of the 3D point within a s-voxel have the same ground truth label, the s-voxel is assigned that same label. Otherwise, the segment is labeled as "NoN" and we don't use it for training phase. The classifier is then trained to distinguish among single-label segments. The classifiers use all of the listed features. Many of these cues can be quickly computed for the s-voxel, since the s-voxel cues provide sufficient statistics. The performance of the classification method is quite good, the result of the classier will be discussed in detail at chapter 5.

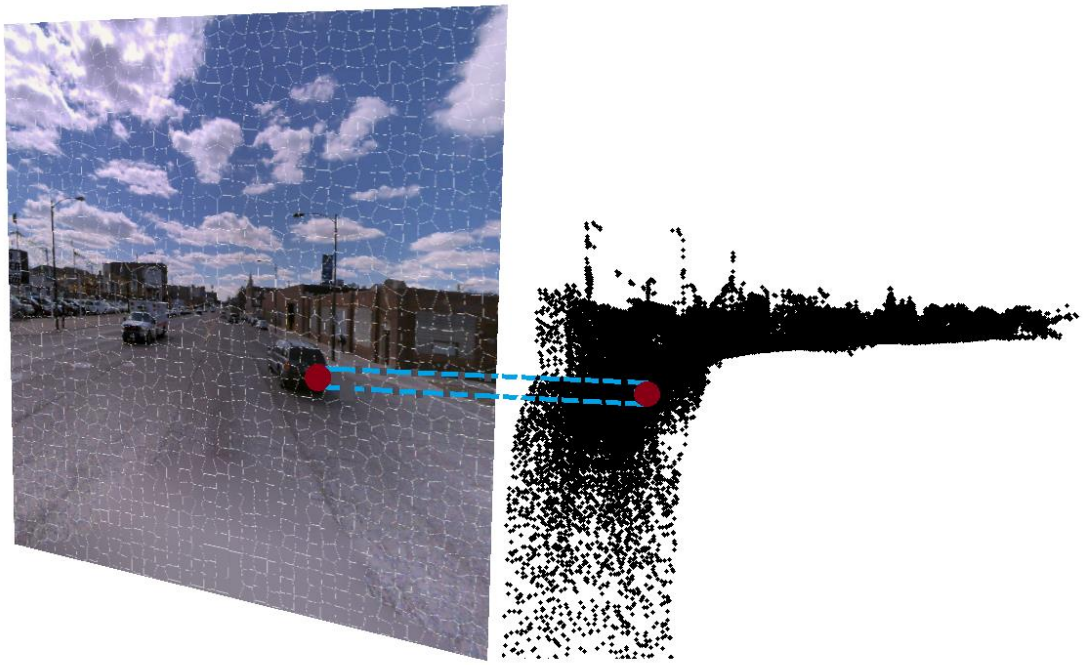


Figure 12. *From 3D world to image plane*

4.6 2D-3D association

In this phase we propose a novel street scene image semantic parsing framework, which takes advantage of 3D labeled point clouds captured by a LiDAR laser scanner. Local 3D geometrical features extracted from s-voxel are classified by trained boosted decision trees and now they are used for labeling corresponding image segments. In contrast to existing image-based scene parsing approaches, the proposed point cloud based approach is robust to varying imaging conditions such as lighting and urban structures.

With the advancement of LiDAR sensors, GPS and IMU devices, large-scale, accurate and dense point cloud can be created and used for 2D scene parsing purpose. There has been a considerable amount of research in registering 2D images with 3D point clouds [73, 74]. Furthermore, there are methods designed for registering point cloud to image using LiDAR intensity [75].

The cubic images and 3D labeled LiDAR point cloud (output of chapter 4.5) are the inputs of the processing step and parsing results are image segments assigned with different class labels. The proposed parsing pipeline starts from aligning 3D LiDAR point cloud with 2D images. Input images are segmented into superpixels to reduce computational complexity and to maintain sharp class boundaries. Each superpixel in 2D image is associated with a collection of labeled LiDAR points, which is assumed to form a planar patch in 3D world. The detailed analysis for each process will be discussed in the following subsections.

4.6.1 Segmenting Images into Superpixels

Without any prior knowledge about how image pixels should be grouped into semantic regions, one commonly used data driven approach segments the input image into homogeneous regions i.e. superpixels based on simple cues such as pixel colors and/or filter responses. The use of superpixels improves the computational efficiency and increases the chance to preserve sharp boundaries between different segments.

In our implementation, we adopt the geometric-flow based technique of Levinshtein [76] to segment images into superpixels with roughly the same size. Sharp image edges are also well preserved by this method. For input images with dimensionality of 2032×2032 pixels, we set the initial number of superpixels as 2500 for each image. See the image in figure 12 as the example of superpixel segmentation results.

4.6.2 LiDAR point cloud to Superpixel

Given a labeled 3D points cloud and one 2D image with known viewing camera pose, the association module described in this section aims to establish correspondences between collections of 3D points and groups of 2D image pixels. In particular, every collection of 3D points is assumed to be sampled from a visible planar 3D object i.e. patch and corresponding 2D projections are confined within a homogenous region i.e. superpixels of the image. While the 3D-2D projection between patches and superpixels is straightforward for known geometrical configurations, it still remains a challenging task to deal with outlier 3D points in a computationally efficient manner. We first review how to project a 3D point on 2D image plane with known viewing camera pose, and then illustrate a method that associates a collection of 3D points with any given superpixel on 2D image.

Given a viewing camera pose i.e. position and orientation, represented, respectively, by T a 3×1 translation vector and R a 3×3 rotation matrix, and a 3D point $M=[X,Y,Z]^t$, expressed in a Euclidean world coordinate system, then the 2D image projection $m_p=[u,v]^t$ of the point M is given by equation 4.7.

$$\widetilde{m}_p = \mathrm{k}\,[\,\mathrm{R}\mid\mathrm{T}\,]\,\widetilde{M} = \mathrm{C}\,\widetilde{M} \qquad (4.7)$$

Where k is an upper triangular 3×3 matrix

$$K = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (4.8)$$

Where f_x and f_y are the focal length in the x and y directions respectively, x0 and y0 are the offsets with respect to the image axes, and $\widetilde{m}_p = [u, v, 1]^t$ and $\widetilde{M} = [x, y, z, 1]^t$ are the homogeneous coordinates of m_p and M.

The motion of a vehicle-based terrestrial mapping system is described in a local coordinate frame. The determination of the position and attitude of the vehicle, or platform, is based on measurements from various sensors attached to the sensor platform on the vehicle, typically a GPS-IMU system. These sensors deliver physical quantities, i.e., accelerations, position and rotation measured within independent frames, and each defined according to the instrument's characteristics.

Global frames are the Earth centered inertial and the Earth centered Earth fixed frames (ECEF that define the position of the MLS on the Earth surface. 3D LiDAR point clouds are often measured in a geographic coordinate system (i.e. longitude, latitude, altitude), therefore, projecting a 3D LiDAR point on 2D image plane involves one more transformation step, namely Geo-to-ECEF (equation 4.9).

$$x = (N + h)cos\phi \cos\lambda$$

$$y = (N + h)cos\phi \sin\lambda \qquad (4.9)$$

$$Z = (N\,(1-e^2) + h)\sin\phi$$

Where ϕ, λ and h are latitude, longitude and height coordinate of 3D point. From WGS-84 the geodetic parameters equal to a= 6378137 and e^2 =0.0067, where N is defined as

$$N = \frac{a}{\sqrt{1-e^2 sin^2\phi}} \qquad (4.10)$$

The relation between Geodetic (ellipsoidal) and ECEF coordinate system are shown in figure 13.

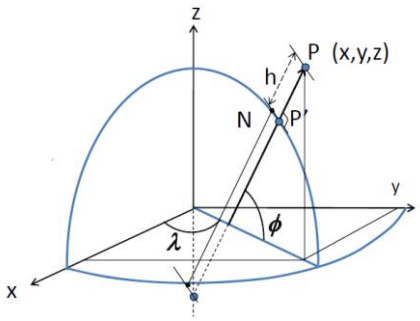


Figure 13. *Coordinate conversion from Geodetic to ECEF*

After this transformation, 3D point are transformed to local coordinate system NED (North=x East=y Down=z) by equation 4.7. Using these necessary transformation step and orthogonal projection, we are able to identify those 3D points that are projected within a specific SP.

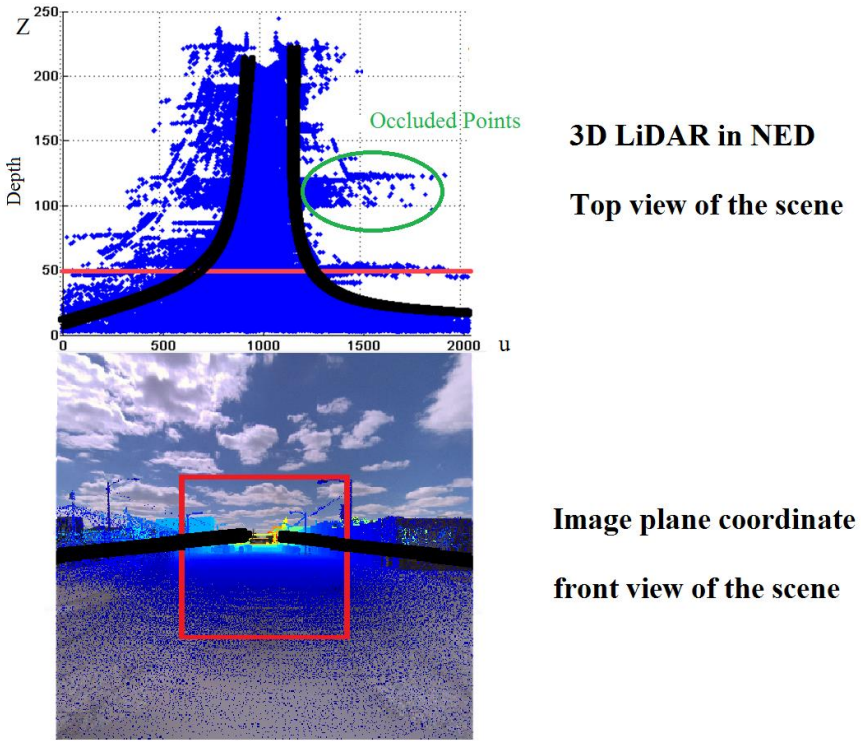


Figure 14. Removing occluded points*. The top image shows 3D LiDAR point cloud in NED system. The occluded points in the one bystreet are shown in a green circle. The Bottom image illustrates camera view of scene, occluded points in the bystreet located in the red square (which corresponding to red line in top image) will be deleted*

To generate projected labeled image for the data collected from the rotating 360 degree laser, we need a projection which can preserve photometric integrity, and provide equal attention to every direction of view. We start by dividing the sphere (representing directions of view) into 6 equal areas which correspond to the faces of an inscribed cube with vertices |x| = |y| = |z|. We generate projection only for front, back and two side views.

Since we assume there is only one dominant 3D patch that associates with the given SP, so outlier 3D points that are far from the patch should be removed. However, such an outlier removal methods have to be repeatedly applied to every superpixel and turns out to be too computationally demanding for our application. In this paper, we instead propose a novel and simple method to remove outlier points for all superpixels in one pass. The proposed method takes advantage of priori knowledge about urban scene environment and assumes that there are building facades along both sides of the street. While this assumption appears to be oversimplified, the method actually performs quite well with urban scenes in our datasets as demonstrated in the experimental results.

Note that we apply the outlier removal for only in front and back view cubic images which contain huge amount of point cloud in their orthogonally projection. The remains two other side view (left and right view, see building facades) cubes images only contain a few number of points. As can be seen in top image in figure 14, as MLS vehicle goes through the path (250 m), the whole point cloud could be seen at front view cubic images.

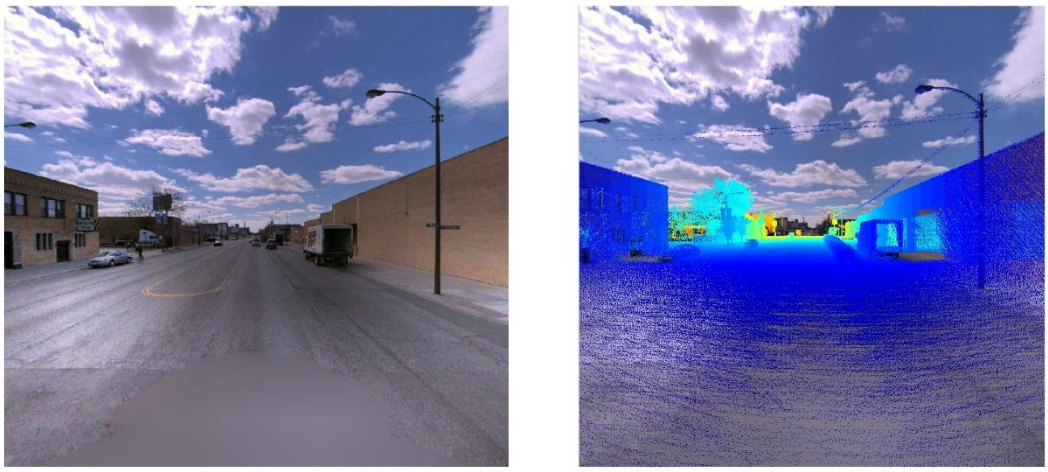


Figure 15. *Depth map generation*

The essence of the method is to fit two hyperbolic curves to 3D points represented in a camera centered two dimensional Z-u plane (see Figure 14 top image). 3D points that are far from camera center and behind these two hyperbolic curves are deeded outliers and are removed. However, points with depth less than 50 meters (see red line) are kept because they play important roles to label road or other near objects. The derivation of hyperbolic curves in this Z-u plane is due to the normalization of homogeneous coordinates or simply:

$$v = \frac{f_y Y}{Z} + y_0 \qquad u = \frac{f_x X}{Z} + x_0 \qquad (4.11)$$

In this case the street width X is assumed constant, u is inversely related to the depth Z, and the collection of aligned points in the 3D world lies between two hyperbolic lines (black lines in figure 14).

Labeled images are generated at cubic image locations. All labeled LiDAR points are converted into local coordinates centered at the panoramic image locations and then mapped onto the superpixels in the four cube faces. If multiple points fall into the same superpixel, the point with minimum distance to the image location (depth map) is chosen to represent the label of superpixel. In the other words the label of 3D point which has the minimum distance to the image location along whole other 3D patch points, assumed as image superpixel label. Figure 15 illustrates the depth map generation properties.

Although we prefer to deem each superpixel belonging to one class, there are some thin structures, such as a pedestrian and sign symbol, which are far from filling the whole superpixel. In this case, a small size superpixel refinement is needed to achieve more accurate results. Superpixel based labeling can increase the chances that the boundaries of different object classes are extracted. In this regard, pixel-wise projection may result in less consistent boundaries. Furthermore using superpixel can reduce the computational

complexity of the system, since by counting each superpixel as one sample, the number of total samples are largely reduced as compared to pixel-wise projection.

5. EXPERIMENTAL RESULT

The LiDAR technology has been used in the remote sensing urban scene understanding by two main technology: Terrestrial Laser Scanning (TLS), useful for large scale buildings survey, roads and vegetation, more detailed but slow in urban surveys in outdoor environments; Mobile Laser Scanning (MLS), less precise than TLS but much more productive since the sensors are mounted on a vehicle; In order to test our algorithm both type of data sets were used:

1. Velodyne LiDAR as TLS dataset [77], only 3D point cloud

2. NAVTAQ True as MLS datasets, contains 3D point cloud + cube images

We train boosted decision tree classifiers with sample 3D features extracted from training s-voxels. Subsequently we test the performance of the trained classifier using separated test samples. The accuracy of each test is evaluated by comparing the ground truth with the scene parsing results. We report global accuracy as the percentage of s-voxel correctly classified, per-class accuracy as the normalized diagonal of the confusion matrix and class average which represents the average value of per class accuracies.

Since no labeled image dataset consisting of corresponding LiDAR point cloud was available (confidential), we created and used labeled dataset of driving sequence from NAVTAQ True, provided by Nokia Research Center, for all 3D and 2D experiment presented in this thesis. To compare our experimental result with other publications we test Velodyne LiDAR dataset which only contains 3D point cloud in local coordinate system and there is not any image to test our 2D image parsing algorithm with.

5.1 Evaluation Using the Velodyne LiDAR Database (3D)

The dataset includes ten high accurate 3D point cloud scenes collected by a Velodyne LiDAR mounted on a vehicle navigating through the Boston area. Each scene is a single rotation of the LIDAR, yielding a point cloud of nearly 70,000 points. Scenes may contain objects including cars, bicycles, buildings, pedestrians and street signs. Finding ground and building points is discussed in Section 4.2 and 4.3, and the recognition accuracy is approximately 98, 4% and 95, 7% respectively. We train our classifier using seven scene datasets, selected randomly, and test on the remaining three scenes.

Table 5.1 presents the confusion matrices between the six classes over all 10 scenes. Our algorithm performs well on most per class accuracies with the heights accuracy 98% for ground and the lowest 72% for sign-symbol. The global accuracy and per-class accuracy are about 94% and 87% respectively.

Table 5.1. *Confusion matrix Velodyne LiDAR Database*

	Tree	Car	Sign	person	Fence	Ground	Building
Tree	0.89	0.00	0.07	0.00	0.04	0.00	0.00
Car	0.03	0.95	0.00	0.00	0.02	0.00	0.00
Sign	0.17	0.00	0.72	0.11	0.00	0.00	0.00
person	0.02	0.00	0.20	0.78	0.00	0.00	0.00
Fence	0.03	0.00	0.00	0.00	0.85	0.00	0.12
Ground	0.00	0.00	0.00	0.00	0.00	0.98	0.02
Building	0.00	0.00	0.00	0.00	0.04	0.00	0.96

Table 5.2. *Comparison of the class accuracy of our approach and Lais approach*

	Tree	Car	Sign	person	Fence	Ground	Building
Lai	0.83	0.91	0.80	0.41	0.61	0.94	0.86
Our	0.89	0.95	0.72	0.88	0.85	0.98	0.95

We also compare our approach to the method described by Lai in [77]. Table 5-2 shows its quantitative testing result. In terms of per class accuracy, we achieve 87% in comparison to 76%. Figure 16 shows some of the qualitative results of the test scene, achieved by our approach. The down image represents classification result of the test scene: non blue points shows incorrectly classified points whereas blue ones are correctly classified points.

As can be seen, most of the objects are classified correctly. The street signs and the car near the facades are not labeled well, since they are not close enough to any exemplar. Furthermore, since each scene is a single rotation of the LIDAR, yielding a cloud of few points, training dataset does not contain enough information to train the classifier well.

Another enormous confusion between classes is found between tree and sign symbol. The signs and some small trees have similar dimension features and it can be difficult even for humans to distinguish them.

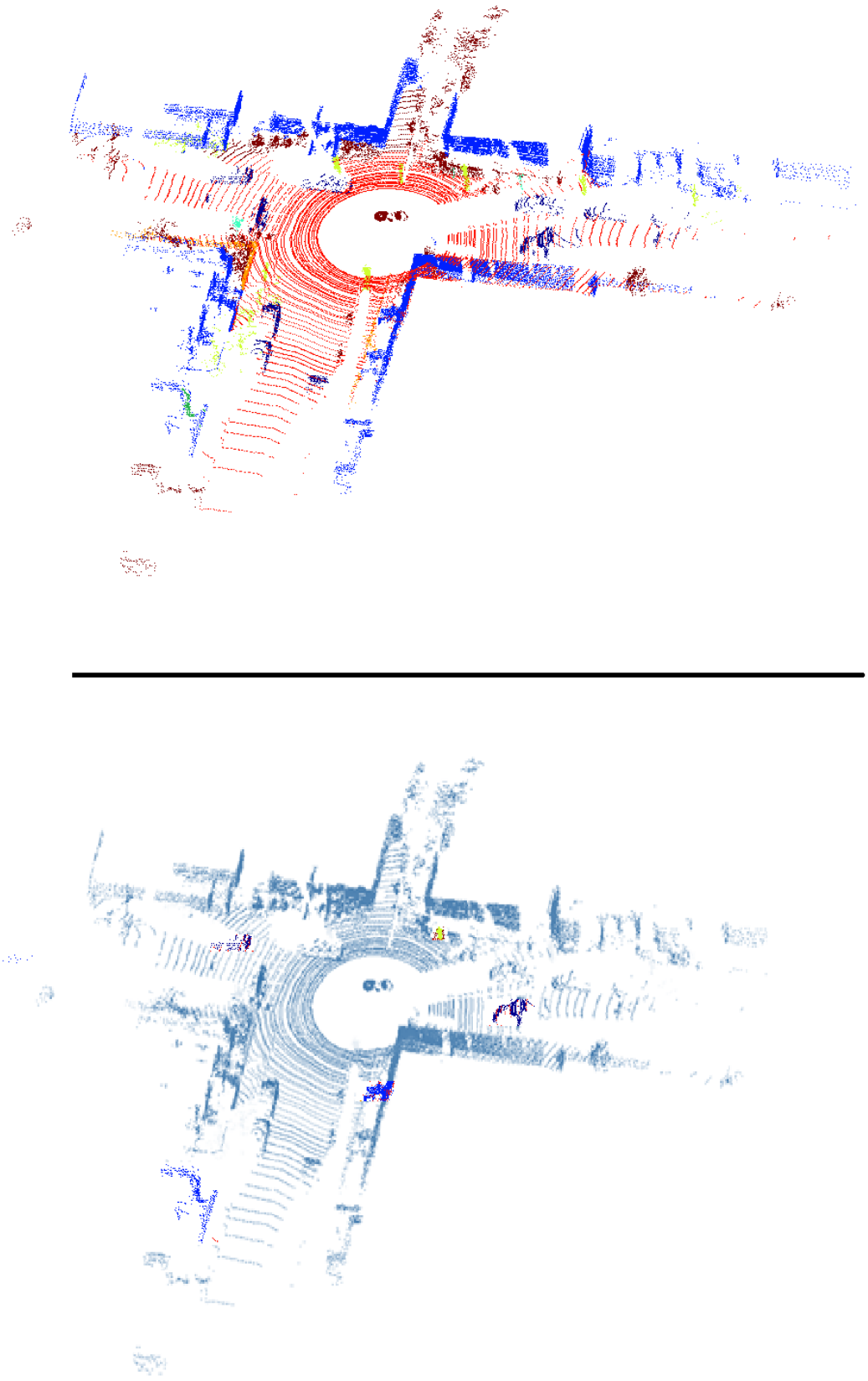


Figure 16. *Top image shows 3D scene object recognition qualitative results, down image represent misclassified 3D points*

***Table 5.3**. NAVTAQ True Dataset properties*

Drive (City)	Helsinki	Chicago	Paris
Approx. Lat, Long	60.1°, 24.9°	41.9°, -87.6°	48.8°, 2.4°
Size of Data (GB)	4.2	6	7.2
Number of Images	100	200	100
Rate (frame/meter)	1/10	1/15	1/10
Temperature	18.5°c	34°c	5°c
Weather Condition	 Sunny	 Partly Cloudy	 Rainy

5.2 Evaluation Using NAVTAQ True datasets

The proposed 3D classification approach is experimented by point cloud captured by NAVTAQ True vehicles. The properties, equipment and sensors embedded on the vehicle are discussed in chapter 2. Since no image labeled dataset (ground truth) consisting of corresponding LiDAR point cloud was available, we created and labeled dataset of driving.

NAVTAQ True datasets contains 3D MLS data. The dataset includes 400 high quality cubic images and corresponding accurate LiDAR point cloud collected from different US and European cities. We selected challenging NAVTAQ drives in different weather conditions (cloudiness, temperature and daytime) and landscapes (shape of buildings, vegetation and vehicles) to evaluate our pipeline precisely (table 5.3). 10 semantic object classes are defined to label the image and corresponding LiDAR dataset: building, tree, sky, car, sign symbol, person, ground, fence, sidewalk and water. It's noteworthy that several objects such as wall sign and wall light are considered as building facades.

Note that some of these classes e.g. building and ground are common objects in the street view images while others such as water, fence and etc. occur less frequently. Furthermore the sky class which covers a large part of whole cubic images surface don't appear in LiDAR dataset. The statistics of occurrences of each class are summarized in figure 17.

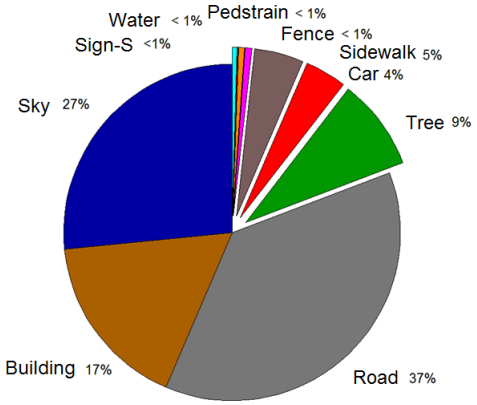


Figure 17. *The statistics of occurrences of each class*

The three NAVTAQ point cloud datasets, collected from US and European countries contains more than 800 million points covering approximately 2 km altogether. These point clouds hold additional information such as RGB color, time step and etc. which is ignored here as our focus remained on using the pure geometry and intensity for the classification of objects.

5.2.1 Evaluation of 3D point cloud classification

The whole three NAVTAQ True data sets are divided into two portions: the training set, and the testing set. The 70% long of each data set are randomly selected and mixed for training of classifier and 30% remained long of point cloud is used for testing. Since all three data sets don't include whole object classes we make experiment based on their common class labels: tree, car, sign symbol, person, bike, ground and building. Table 5.4 shows the quantities results achieved by our approach.

Table 5.4*. Confusion matrix of NAVTAQ True Database (3D point cloud)*

	Tree	Car	Sign	person	Bike	Ground	Building
Tree	0.75	0.07	0.10	0.00	0.00	0.00	0.08
Car	0.11	0.73	0.00	0.00	0.05	0.00	0.11
Sign	0.09	0.00	0.78	0.13	0.00	0.00	0.00
person	0.07	0.00	0.21	0.58	0.14	0.00	0.00
Bike	0.03	0.00	0.00	0.04	0.81	0.00	0.12
Ground	0.00	0.00	0.00	0.00	0.00	0.97	0.03
Building	0.05	0.00	0.00	0.00	0.04	0.00	0.95

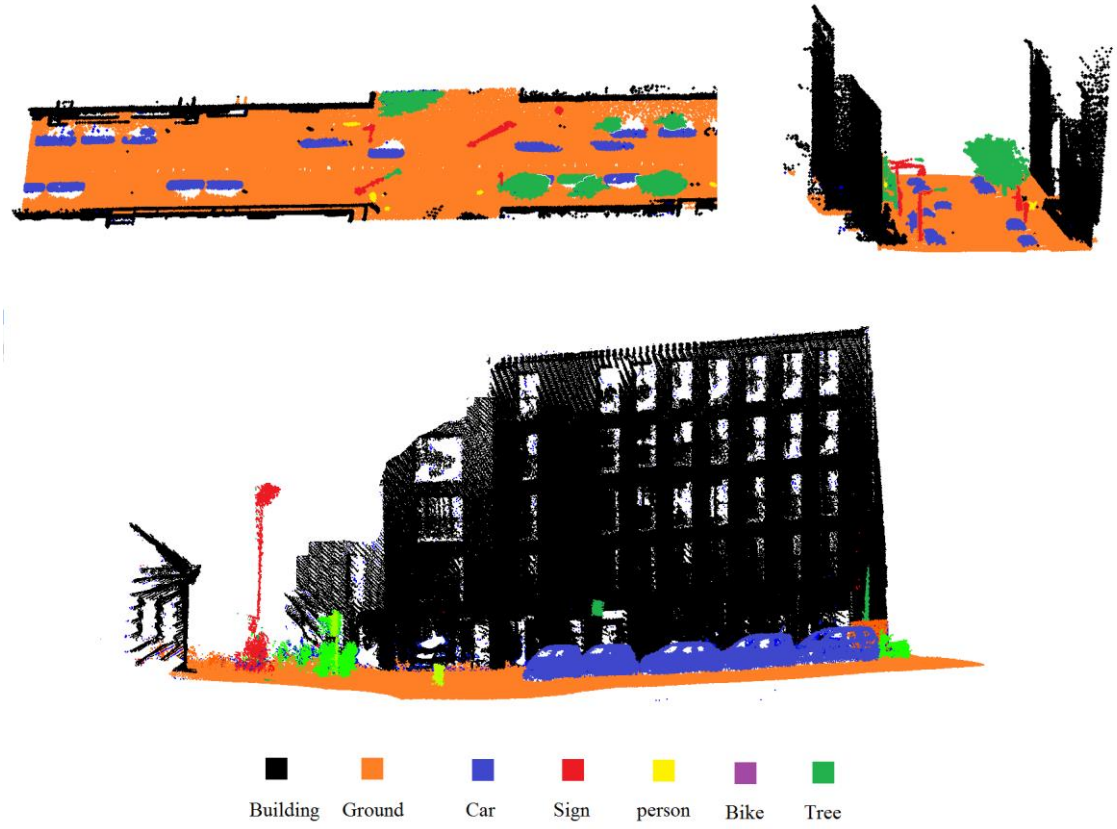


Figure 18. *3D Scene object recognition qualitative results in different view (point cloud)*

Comparing to Terrestrial Laser Scanning, our results are not as good as in shown in Table 5.1. Since mixing three data sets captured from different cities poses serious challenges to the parsing pipeline. Furthermore, Moving objects are even harder to reconstruct based solely on MLS LiDAR data. As these objects (typically vehicles, people) are moving through the scene, which make them appear like a long-drawn shadow in the registered MLS point cloud. The long shadow artifact is not appear in TLS system because in which we face to one point as exposure point to scan the street objects. Figure 18 shows some of the qualitative results of the test scene.

5.2.2 Evaluation of Image parsing based on 3D LiDAR point classification (2D-3D association)

For evaluation of our pipeline based on the given labeled 3D points cloud and 2D cubic images with known viewing camera pose, the association module described in the section 4.6, we have done three different experiments: Direct training and testing, mixed training and testing and cross training and testing. These three experiments have been done to evaluate independency of our algorithm performances in different training-testing classification conditions. In whole three experiments we train boosted decision tree classifiers with sample 3D features extracted from the 3D point cloud projected to image plane.

Subsequently, we test the performance of the trained classifier and proposed algorithm using separated test point cloud. The same tests are applied to three different urban areas.

The accuracy of each test is computed by comparing the 2D ground truth with the scene paring results. We report global accuracy as the percentage of superpixels correctly classified, per-class accuracy (the normalized diagonal of the confusion matrix) and class average which represents the average value of per-class accuracies. Since in each experiment, dataset randomly have been divided to two groups of training and testing categories we repeated each experiment five times and the average of resulted experiment represented as the final accuracy.

Direct training and testing: We randomly split each city dataset into two groups in such a way that 70 percent of the images are used for training the classifier and the remaining 30 percent for testing. Table 5.5 shows the confusion matrixes for different experiments in three cities. As can be seen, some classes in Chicago and Helsinki experiments have not been labeled because there are no sufficient samples for those classes. Our algorithm performs well on most per class accuracies, with the highest accuracy 99% achieved for the sky in Chicago and the lowest 32% for sign-symbol in Paris. The average of the global accuracy for three direct experiments is about 88 %.

Table 5.5.1 Confusion matric for direct classification in Chicago

Chicago	Sky	Building	Road	Tree	Car	Sidewalk	Sign- S	Fence
Sky	99	1	0	0	0	0	0	0
Building	12	84	0	2	0	1	0	1
Road	1	0	97	0	0	1	0	0
Tree	10	32	0	57	0	0	0	1
Car	5	10	24	0	46	13	0	2
Sidewalk	3	13	7	0	10	67	0	0
Sign- S	5	14	0	6	0	34	41	0
Fence	7	40	0	1	4	1	0	47

Table 5.5.2 Confusion matric for direct classification in Paris

Paris	Sky	Building	Road	Tree	Car	Sidewalk	Sign	person	Water
Sky	75	4	21	0	0	0	0	0	0
Building	5	90	0	3	0	1	0	0	1
Road	1	0	91	0	2	6	0	0	0
Tree	5	2	0	88	0	5	0	0	0
Car	2	3	55	0	33	7	0	0	0
Sidewalk	2	1	3	1	1	91	0	0	1
Sign	5	18	14	10	0	25	32	0	6
person	16	24	0	4	0	0	0	47	9
Water	48	5	0	3	0	3	0	0	41

Table 5.5.3 Confusion matric for direct classification in Helsinki

Helsinki	Sky	Building	Road	Tree	Car	Sidewalk
Sky	95	4	0	1	0	0
Building	4	88	0	7	0	1
Road	1	0	96	0	2	1
Tree	1	25	0	74	0	0
Car	10	4	10	0	64	12
Sidewalk	2	15	0	0	26	58

Mixed training and testing: The whole 400 images collected from three cities are randomly mixed and then split into 300 for training and 100 for testing. The mixed classification confusion matrix is shown in table 5.6. It should be noted here that some of the classes have a distinctive geometry and can be classified more easily (e.g., sky and road) whereas others have similar geometrical features (e.g., Fence and building).

Table 5.6: *image parsing statistical results, confusion matrix for mixed classification*

mixed	Sky	Build-ing	Road	Tree	Car	Side-walk	Sign-S	Fence	per-son	Wa-ter
Sky	96	2	0	2	0	0	0	0	0	0
Build-ing	4	90	0	3	0	2	0	1	0	0
Road	2	0	96	0	1	1	0	0	0	0
Tree	6	17	0	74	0	3	0	0	0	0
Car	5	11	35	1	35	11	0	2	0	0
Side-walk	2	4	12	1	4	77	0	0	0	0
Sign- S	8	2	5	4	3	60	17	0	0	1
Fence	5	37	0	3	4	1	0	49	0	1
person	10	34	1	3	3	21	0	0	22	6
Water	48	6	1	5	1	5	0	1	0	33

Mixing images from different cities poses serious challenges to the parsing pipeline, which is reflected by the decrease in the class average accuracy (down to 59%). Nevertheless, it seems our system generalizes well to different city scenes and the comparable global accuracy 88% is still maintained.

Cross training and testing: The idea of cross training and testing is to challenge the system with training and testing images taken from different cities (table 5.7). As expected, our method works well when training in Helsinki and testing in Chicago (79 % global and 52 % class average accuracy) and vice versa (69% global, 42% class average). Comparing to other cross experiments, Chicago and Helsinki cross experiments represent

best parsing accuracy because as discussed earlier there are more similar classes compared to Paris which contains major water in its scene.

Table 5.7: *Compares global and class average accuracy in whole different experiments*

Experiments\Results	Global Accuracy	Class average Accuracy
Direct (Helsinki)	86 %	79 %
Direct (Chicago)	93 %	67 %
Direct (Paris)	85 %	65 %
Mixed	88 %	59 %
Cross (Helsinki-Chicago)	79 %	52 %
Cross (Chicago-Helsinki)	69 %	42 %
Cross (Helsinki-Paris)	59 %	36 %
Cross (Paris-Helsinki)	64 %	41 %
Cross (Chicago-Paris)	61 %	37 %
Cross (Paris-Chicago)	68 %	45 %

Applying SP based segmentation to relatively small classes such as pedestrian and sign-symbol often leads to insufficient number of training samples, and hence, low classification accuracies. The plot in Figure 19 illustrates the qualitative comparison between per class accuracy according to their distribution in our datasets. It should be noted that sky, building, road and tree were well recognized in the street scene (all are over 70%). On the other hand, cars and pedestrian have less than 10% accuracies because these classes occur very rarely in the test images. One possible remedy is to obtain the bounding boxes of these objects with a more suitable technique, e.g. part-based object detector.

Our system takes advantage of geographical and intensity statistics information of LiDAR point clouds. The bar chart in figure 19 shows that using intensity feature improves classification accuracies, to various extents, for objects e.g. building, car, and signs-symbol and pedestrian. There also seems a discernible increase in its effectiveness as objects become closer to the laser scanner.

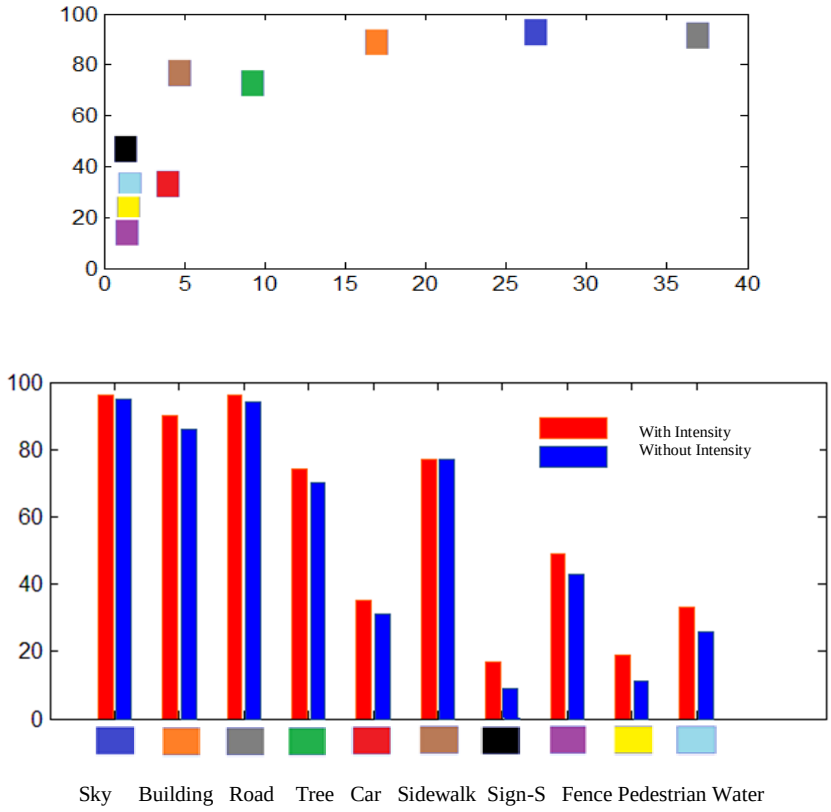


Figure 19. *Top plot compares the accuracy of mixed classification based on distribution of existing data (Per Class accuracy in percent). Bottom bar graph shows the impact of intensity feature in mixed training-testing experiment.*

6. CONCLUSIONS

We have proposed a novel and comprehensive framework for semantic parsing of street view 3D MLS and TLS point cloud based on geometrical features. First, ground are segmented using a heuristic approach based on the assumption of constant slope group plane. We used the RANSAC algorithm to fit a surface plane to the LiDAR 3D points. The fitted surface consists of slightly skewed cells. This step is intended to provide an accurate representation of the ideal road surface. Second, building points are then extracted by tracing contours of projections of 3D points onto the x - y plane. Using this segmentation huge amount of data (more than 75% of points) are labeled, and only small amount of point cloud which have complex shape remained to be segmented.

The connected component classification was used to ensure each object is connected as one component. During the offline training phase 3D features are extracted at s-voxel level and are used to train boosted decision trees classifier. Because the properties of s-voxels are constant mainly over the whole points and these properties are then used for classification, their size impacts the classification process. With smaller voxel size the segmentation and classification results are improved but the computational cost increases.

For new scene, the same unsupervised ground and building detection are applied and geometrical features are extracted and semantic labels are assigned to corresponding point cloud area. The proposed two-stage method requires only small amount of time for training while the classification accuracy is robust to different types of LiDAR point clouds acquisition methods.

Finally, we introduced a method for utilizing an existing 3D classification approach to improve and generate accurate image parsing. Given a labeled 3D points cloud and 2D image with known viewing camera pose, the proposed association module aligned collections of 3D points to the groups of 2D image pixel to parsing 2D cubic images. One noticeable advantage of our method is the robustness to different lighting condition, shadows and city landscape.

The feature extraction is one of the main work of this research and decided features are extremely important for the classification. Although whole 10 features have been extracted, some of them are rarely used in the classification while some of them contribute to the error. Furthermore, by using intensity information from LiDAR data the robustness of classifier is increased for certain object classes.

There are three main errors sources: error from method, assessment and dataset. The error from the ground segmentation, connected component analysis, future extraction and classification have been discussed and the errors have been reduced as much as possible. The manually labeled reference dataset includes some defect over the recognition experiment. The ground truth do not have sufficient reference component for each class and there might be some mistakes during the recognition of the object visually. The accuracy of our approach depends on the quality of LiDAR resolution and accuracy. While we have applied state-of-the-art algorithms to object recognition, the quality of LiDAR is still quite fragile. Our next step is to apply more accurate MLS and ALS scan data that acquired at the same time. We expect to see large performance improvement with better this type of input data. In addition, one future area of work is the bias against less frequently appearing objects, such as thin column poles and pedestrians. This is mainly due to lack of sufficient training examples, which naturally lead to a less statistically significant labeling for objects in these classes. Another interesting future work will consider real-time implementation for prediction, better handling of small objects, and extend the method to more general contexts beyond street view.

This algorithm was done based on characteristics of objects in urban street scenes but it can be done for other environment like faubourg, country or even indoor spaces by inspecting characteristics of its objects and appropriate definition of constraints. As it can be seen in the figure 20, successful point cloud classification and alignment have been done accurately. To our best knowledge, no existing methods have demonstrated the robustness with respect to variety in LiDAR point data. We have processed data on a large scale and achieved satisfactory accuracy and performance.

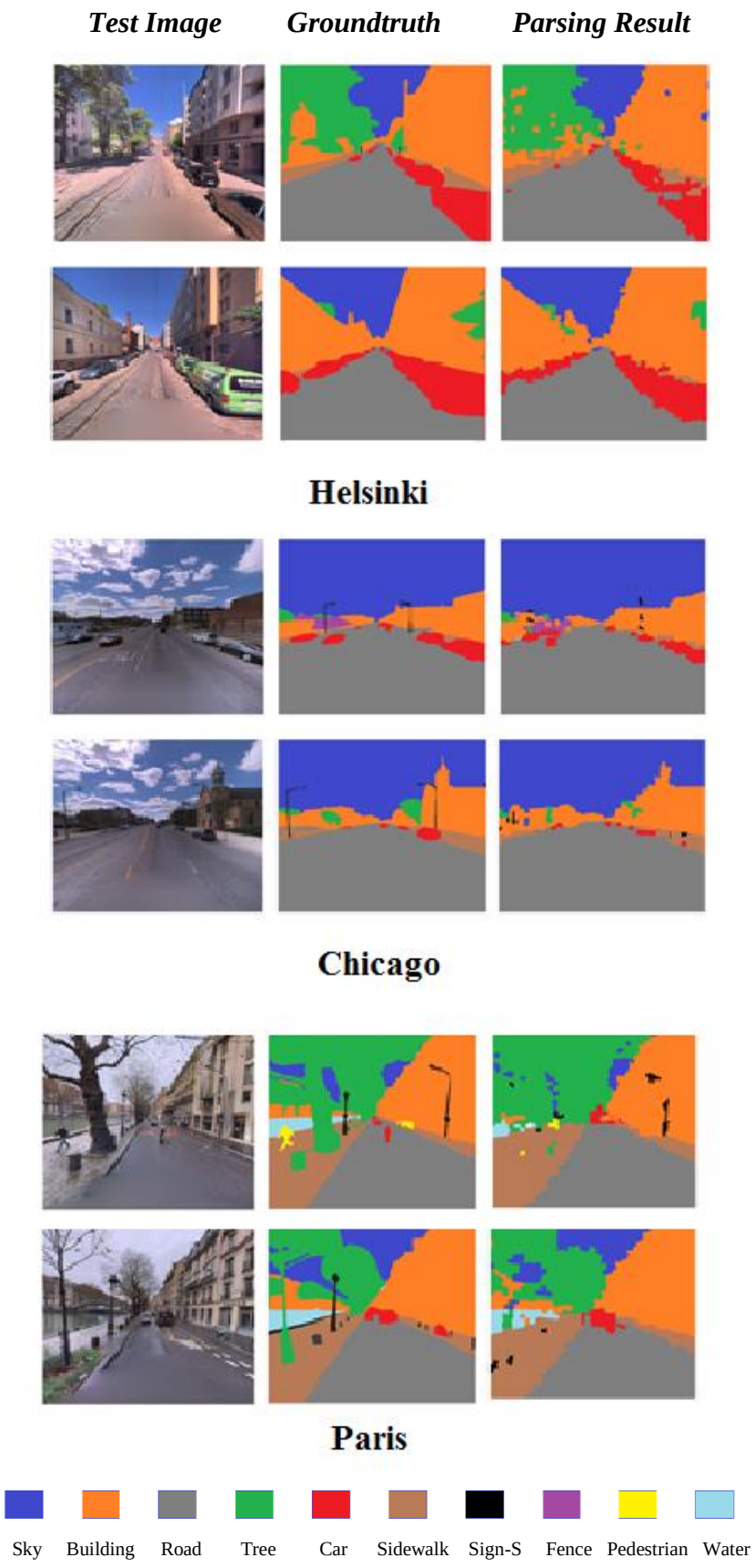


Figure 20. *Scene parsing qualitative results. (Left to right): test image, ground truth (manually labeled image), parsing result in Helsinki, Chicago and Paris cites respectively from top to bottom*

References

1) G. F. Marshal, G. E. Sautz, Handbook of optical and laser scanning, CRC Press 2011.
2) G. V. Vosselman, H. G Maas, and others, Airborne and terrestrial laser scanning. Scotland: Whittles, 2010.
3) A. G. Zavodny, Change detection in lidar scans of urban environments, University of notre dame: Doctoral Dissertation, 2012.
4) A. Wehr, U. Lohr, Airborne laser scanning introduction and overview. ISPRS Journal of Photogrammetry and Remote Sensing 54: 68–82, 1999.
5) L. Matikainen, J. Hyyppa, and H. Hyyppä, Automatic detection of buildings from laser scanner data for map updating, ISPRS Commission III. Workshop 3-d reconstruction from airborne laserscanner and InSAR data, 2003.
6) P. Krishnamoorthy, K. L. Boyer, and P. J. Flynn, Robust detection of buildings in digital surface models, Proceedings 16th International Conference on Pattern Recognition, Quebec City, Que., Canada, pp. 159-63, 2002.
7) D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, Discriminative learning of Markov random fields for segmentation of 3D scan data, Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, San Diego, CA, United States, pp. 169-176, 2005.
8) S. Matzka, Y. R. Petillot, and A. M. Wallace, Determining efficient scan-patterns for 3-D object recognition using spin images, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Heidelberg, D-69121, Germany, pp. 559-570, 2007.
9) R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, Matching 3D models with shape distributions, Proceedings International Conference on Shape Modeling and Applications, Los Alamitos, CA, USA, pp. 154-66, 2001.
10) R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, Shape distributions, CM Transactions on Graphics, vol. 21, pp. 807-832, 2002.
11) A. S. Mian, M. Bennamoun, and R. Owens, Matching tensors for automatic correspondence and registration, Computer Vision - ECCV 2004. 8th European Conference on Computer Vision. Proceedings (Lecture Notes in Comput. Sci. Vol.3022), Berlin, Germany, pp. 495-505, 2004.
12) W. Zhaohui, W. Yueming, and P. Gang, 3D face recognition using local shape map,"2004 International Conference on Image Processing (ICIP) (IEEE Cat. No.04CH37580), Piscataway, NJ, USA, pp. 2003-6, 2004.
13) T. W. Way, H.-P. Chan, M. M. Goodsitt, B. Sahiner, L. M. Hadjiiski, C. Zhou, and A. Chughtai, Effect of CT scanning parameters on volumetric measurements of pulmonary nodules by 3D active contour segmentation: A phantom study, Physics in Medicine and Biology, vol. 53, pp. 1295-1312, 2008.
14) D. D. Lichti, Spectral filtering and classification of terrestrial laser scanner point clouds, Photogrammetric Record, vol. 20, pp. 218-240, 2005.

15) F. Rottensteiner, Automatic generation of high-quality building models from lidar data, IEEE Computer Graphics and Applications, vol. 23, pp. 42-50, 2003.
16) P. Shi, Knowledge based building façade reconstruction laser point clouds images, PhD project, 2009.
17) J. Dash, E. Steinle, R.P Singh and H. P Bähr, Automatic building extraction from laser scanning data: an input tool for disaster management. Advances in Space Research, 33, (3), 317-322, 2004.
18)) J. A. González, B. Rodríguez, Terrestrial laser scanning intensity data applied to damage detection for historical buildings, Journal of Archaeological Science, 2010.
19) X. Hu, C.V. Tao, Y. Hu, Automatic Road Extraction from Dense Urban Area by Integrated Processing of High Resolution Imagery and Lidar Data, Processing of High Resolution Imagery and LIDAR Data, p 320-324, 2004.
20) P. Shi, R. Martin, V. George, O. E. Sander, Recognizing basic structures from mobile laser scanning data for road inventory studies, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 66, Issue 6, p. S28-S39. 2011.
21) J. R. Rosell, J. Llorens, R. Sanz, J Arno, M. Ribes-Dasi, J. Masip, A Escolà, F. Camp, F. Solanelles, F. Gràcia, et al, Obtaining the three-dimensional structure of tree orchards from remote 2D terrestrial LIDAR scanning. Agr. Forest. Meteorol.149:1505–1515, 2009.
22) P. Babahajiani, L. Fan, M. Gabbouj, Object Recognition in 3D Point Cloud of Urban Street Scene, IEEE Asian Conference on Computer Vision (ACCV), Singapore 2014.
23) C. Liu, J. Yuen, Torralba, nonparametric scene parsing: Label transfer via dense scene alignment. In: Computer Vision and Pattern Recognition, CVPR 2009. IEEE Conference on, IEEE 2009.
24) G. Csurka, F. Perronnin, A simple high performance approach to semantic segmentation. In: BMVC, 2008.
25) D. Hoiem, A. Efros, M. Hebert, Recovering surface layout from an image. International Journal of Computer Vision 75, 2007.
26) G. Floros, B. Leibe, Joint 2d-3d temporally consistent semantic segmentation of street scenes. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 2823-2830.
27) G. Zhang, J. Jia, T. Wong, H. Bao, Consistent depth maps recovery from a video sequence, Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 (2009) 974-988.
28) W. L Lu, K. P. Murphy, and J. J. Little, A. Shefier, H. Fu, A hybrid conditional random field for estimating the underlying ground surface from airborne lidar data, Geoscience and Remote Sensing, IEEE Transactions on 47 (2009) 2913-2922.
29) J, Hern_andez, B. Marcotegui, Filtering of artifacts and pavement segmentation from mobile lidar data, In: ISPRS Workshop Laserscanning 2009.
30) Y. Zhou, Y. Yu, G. Lu, S. Du, Super-segments based classification of 3D urban street scenes. Int J Adv Robotic Sy 9, 2012.

31) A. Johnson, Spin-Images: A Representation for 3-D Surface Matching. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, 1997.
32) M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3D shape descriptors. In: Symposium on geometry processing. Volume 6, 2003.
33) J. Sun, M. Ovsjanikov, L. Guibas, A concise and provably informative multi-scale signature based on heat diffusion. In: Computer Graphics Forum. Volume 28, Wiley Online Library (2009) 1383-1392.
34) R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions. ACM Transactions on Graphics (TOG) 21 (2002) 807-832.
35) J. Knopp, M. Prasad, L. Van Gool, Orientation invariant 3D object classification using hough transform based methods. In: Proceedings of the ACM workshop on 3D object retrieval, ACM (2010) 15-20.
36) T. Rabbani, D. Van, F. Heuvel, G. Vosselmann, Segmentation of point clouds using smoothness constraint. Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci. 2006, 36, 248–253.
37) F. Moosmann, O. Pink, C. Stiller, Segmentation of 3D Lidar Data in non-flat Urban Environments Using a Local Convexity Criterion. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Shaanxi, China, 3–5 June 2009; pp. 215–220.
38) A. Golovinskiy, T. Funkhouser, Min-Cut Based Segmentation of Point Clouds. In Proceedings of the IEEE Workshop on Search in 3D and Video (S3DV) at ICCV, Nara, Japan, 29 September–2 October 2009; pp. 39–46.
39) C. T. Zahn, Graph-theoretic methods for detecting and describing gestalt clusters. IEEE Transactions on Computing, 20:68–86, 1971.
40) X. Zhu, H. Zhao, Y. Liu, T. Zhao, H. Zha, Segmentation and Classification of Range Image from an Intelligent Vehicle in Urban Environment. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010; pp. 1457–1462.
41) J. Strom, A. Richardson, E. Olson, Graph-Based Segmentation for Colored 3D Laser Point Clouds. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010; pp. 2131–2136.
42) F. Pauling, M. Bosse, R. Zlot, Automatic Segmentation of 3D Laser Point Clouds by Ellipsoidal Region Growing. In Proceedings of the Australasian Conference on Robotics & Automation, Sydney, Australia, 2–4 December, p. 10, 2009.
43) D. Anguelov, B, Taskar, V, Chatalbashev, D. Koller, D. Gupta, G. Heitz, A. Ng, Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data. In Proceedings of IEEE Computer Society Conference on the Computer Vision and Pattern Recognition, Los Alamitos, CA, USA, Volume 2, pp. 169–176, 20–26 June 2005.
44) E. Lim, D. Suter, Conditional Random Field for 3D Point Clouds with Adaptive Data Reduction. In Proceedings of the International Conference on Cyberworlds, Hannover, Germany, pp. 404–408, 24–26 October 2007.

45) W. L. Lu, K. Okuma, J. J. Little, A hybrid conditional random field for estimating the underlying ground surface from airborne LiDAR data. IEEE Trans. Geosci. Remote Sens. 2009.
46) G. Vosselman, P. Kessels, B. Gorte, The utilization of airborne laser scanning for mapping. Int. J. Appl. Earth Obs. Geoinf. 2005.
47) S. Pu, G. Vosselman, Building facade reconstruction by fusing terrestrial laser points and images. Sensors 2009.
48) O. Hadjiliadis, I. Stamos, Sequential Classification in Point Clouds of Urban Scenes. In Proceedings of the 3DPVT, Paris, France, 17–20 May 2010.
49) T. Funkhouser, P. Min, M.Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3d models. In ACM Trans. on Graphics, pages 22:83–105, January 2003.
50) R.Osada, T.Funkhouser, B. Chayelle, and D. Dobkin. Matching 3d models with shape distributions. In Shape Modeling International, May 2001.
51) D. Huber, A. Kapuria, R. Donamukkala, and M. Hebert. Parts-based 3d object classification. In CVPR, 2004.
52) S. Ruiz-Correa, L. Shapiro, M. Meila, and G.Berson. Discriminating deformable shape classes. In NIPS, 2004
53) F. Han, Z. Tu, and S.-C. Zhu. Range image segmentation by an efficient jump-diffusion method. In IEEE Trans. on Pattern Analysis and Machine Intelligence, Sept 2004
54) A. Halma, F. Ter Haar, E. Bovenkamp, P. Eendebak, Single Spin Image-ICP Matching for Efficient 3D Object Recognition. In Proceedings of the ACMWork-shop on 3D Object Retrieval (3DOR ’10), Norrk¨oping, Sweden, pp. 21–26, 2 May 2010.
55) R. Rusu, G. Bradski, R. Thibaux, J. Hsu, Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, pp. 2155–2162, 18–22 October 2010.
56) Y. Liu, H. Zha, H. Qin, Shape Topics-A Compact Representation and New Algo-rithms for 3D Partial Shape Retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; Volume 2, pp. 2025–2032, 2006.
57) E. H. Lim, D. Suter, Multi-scale Conditional Random Fields for Over-Segmented Irregular 3D Point Clouds Classification. In Proceedings of the Computer Vision and Pattern Recognition Workshop, Anchorage, AK, USA, 23–28 June 2008; pp. 1–7, 2008.
58) J. Lam, K. Kusevic, P. Mrstik, R. Harrap, Greenspan, M. Urban Scene Extraction from Mobile Ground Based LiDAR Data. In Proceedings of the International Symposium on 3D Data Processing Visualization and Transmission, Paris, France, 17–20 May 2010; p. 8, 2010.
59) B. Douillard, A. Brooks, F. Ramos, A 3D Laser and Vision Based Classifier. In Proceedings of the 5th International Conference on Intelligent Sensors, Sensor

Networks and Information Processing (ISSNIP), Melbourne, Australia, 7–10 December 2009; p. 6, 2009.

60) A. Golovinskiy, V. G. Kim, T. Funkhouser, Shape-based recognition of 3D point clouds in urban environments, Computer Vision, 2009 IEEE 12th International Conference on, 2154-2161, 2009.
61) K. Khoshelham, S. O. Elberink, Role of dimensionality reduction in segment-based classification of damaged building roofs in airborne laser scanning data. Proceedings of the 4th GEOBIA, p.372, May 7-9, 2012
62) A. K. Jain, R. P. W. Duin, J. Mao, Statistical pattern recognition: A review, Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (1), 4-37, 2000
63) R. B. Rusu, S. Cousins, 3D is here: Point Cloud Library (PCL), IEEE International Conference on Robotics and Automation (ICRA), 2011.

64) M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24 (6), 381–395, 1981.
65) T. Pavlidis, Algorithms for graphics and image processing. Computer science press, 1982.
66) Y. Zhou, Y. Yu, G. Lu, S. Du, Super-segments based classification of 3d urban street scenes. Int J Adv Robotic Sy 9 (2012)
67) K. Klasing, D. Althoff, D. Wollherr, M. Buss, Comparison of surface normal estimation methods for range sensing applications. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE (2009) 3206–3211, 2009
68) C. Zhang, L. Wang, R. Yang, Semantic segmentation of urban scenes using dense depth maps. In: Computer Vision–ECCV 2010, Springer 2010.
69) P. Babahajiani, L. Fan, M. Gabbouj, Semantic parsing of street scene images using 3D lidar point cloud. Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops 13, 2013.
70) J. Xiao, L. Quan, Multiple view semantic segmentation for street view images. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE 2009.
71) M. Collins, R. E. Schapire, Y. Singer, Logistic regression, adaboost and bregman distances. Machine Learning 48, 2002.
72) D. Hoiem, A. A. Efros, M. Hebert, Recovering surface layout from an image. International Journal of Computer Vision 75, 2007.
73) L. Liu, and I. Stamos, "Automatic 3d to 2d Registration for the Photorealistic Rendering of Urban Scenes," In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, p. 137–143, 2005
74) R. Wang, F.P. Ferrie, and J. Macfarlane, "Automatic Registration of Mobile LiDAR and Spherical Panoramas," CVPR Workshops, pp. 33-40, 2012
75) R. Wang, J. Bach, J. Macfarlane, and F.P. Ferrie, "A New Upsampling Method for Mobile LiDAR Data," WACV, pp. 17- 24, 2012

76) A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast Superpixels Using Geometric Flows," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 31, no. 12, p. 2290–2297, 2009.
77) K. Lai, D. Fox, Object recognition in 3d point clouds using web data and domain adaptation. The International Journal of Robotics Research 29, 2010.